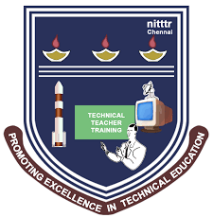
DravidianLangTech 2026

Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages

Proceedings of the Workshop

July 3, 2026

The DravidianLangTech organizers gratefully acknowledge the support from the following sponsors.



©2026 Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
317 Sidney Baker St. S
Suite 400 - 134
Kerrville, TX 78028
USA
Tel: +1-855-225-1962
acl@aclweb.org

ISBN 979-8-89176-401-9

Introduction

We are excited to welcome you to the sixth workshop on Speech, Vision, and Language Technologies for Dravidian Languages (DravidianLangTech 2026), the 64th Annual meeting of the Association for Computational Linguistics (ACL 2026). This year, the workshop will be held in a virtual format (only online) via Underline on July 4th, 2026, alongside ACL 2026, which will take place from July 02 to 07, 2026. With the rapid advancement of technology, digital communication has become a central part of daily life. While many globally dominant languages have successfully transitioned into the digital era, numerous regional and low-resource languages continue to face significant technological challenges. A prominent example of such a language family is the Dravidian language family, which remains underrepresented in the domains of speech and natural language processing (NLP). The DravidianLangTech-2026 workshop series aims to address the technological marginalization of Dravidian languages by fostering research and development in computational linguistics, speech processing, and NLP specific to these languages. By building inclusive language technologies, the goal is to ensure equitable access to digital information and communication tools for monolingual speakers of Dravidian languages. These workshops represent an important step toward preserving linguistic diversity and preventing the digital extinction of these historically and culturally significant languages. This will be the sixth workshop on speech and language technologies for Dravidian languages, continuing our mission to advance technological solutions and promote linguistic inclusivity. The workshop received a total of 123 active submissions, with an additional 4 withdrawn and 7 desk rejected. Reviewer recruitment was highly successful with 195 reviewers accepting invitations. Of the 704 assigned reviews, 379 were completed, achieving a review submission rate of 53.84%. Additionally, 46.67% of reviewers (91 out of 195) fulfilled all their assigned tasks. Among the submissions, 69.11% (85 out of 123) received at least three reviews, reflecting a comprehensive evaluation process. Decisions were made for 100.00% of submissions (123 out of 123), resulting in an overall acceptance rate of 57.72% (71 papers). This included 17 papers (13.82%) accepted for oral presentations and 54 papers (43.90%) accepted for poster presentations. The remaining 52 papers (42.28%) were rejected after review. These statistics highlight a rigorous yet inclusive selection process supported by dedicated reviewers and a commitment to academic excellence.

Program Committee

Program Chairs

Bharathi Raja Chakravarthi, University of Galway, Ireland
Ruba Priyadharshini, The Gandhigram Rural Institute - Deemed University, Tamil Nadu, India
Anand Kumar Madasamy, National Institute of Technology Karnataka, India
Sajeetha Thavareesan, Eastern University, Sri Lanka
Saranya Rajiakodi, Central University of Tamil Nadu, India
Subalalitha Chinnaudayar Navaneethakrishnan, SRM Institute of Science and Technology, Tamil Nadu, India
Dhivya Chinnappa, Legion Intelligence, Texas, United States
Balasubramanian Palani, Indian Institute of Information Technology (IIIT) Kottayam, India
Malliga Subramanian, Kongu Engineering College, Tamil Nadu, India
Kogilavani Shanmugavadivel, Kongu Engineering College, Tamil Nadu, India
Ratnavel Rajalakshmi, Vellore Institute of Technology, Tamil Nadu, India

Publication Chairs

Shunmuga Priya Muthusamy Chinnan, Data Science Institute, University of Galway, Ireland
Premjith B, Amrita School of Artificial Intelligence, Amrita Vishwa Vidyapeetham, Coimbatore, India
Prasanna Kumar Kumaresan, Data Science Institute, University of Galway, Ireland

Best Reviewers

Saurabh Garg, Amazon
Surangika Ranathunga, Massey University
Enzo Doyen, LiLPa, Université de Strasbourg

Program Committee

Abhay Goyal, Missouri University of Science and Technology
Abhishake Reddy Onteddu, University of Central Missouri
Adeep Hande, Comcast Applied AI
Adnan Faisal, Chittagong University of Engineering and Technology
Ahamed Rameez Mohamed Nizzad, Zayed University
Ajees A P, Cochin University of Science and Technology, Dhirubhai Ambani Institute Of Information and Communication Technology
Aleksander Wieczorek, University of Basel
Amritha Prabakaran, National University of Ireland, Galway
Anagha Ananth, University of California, Berkeley
Anbukkarasi Sampath, Manipal University
Andrew Li, Lake Washington School District
Anjishnu Mukherjee, George Mason University
Arunaggiri Pandian Karunanidhi, Micron Technology
Arup Baruah, Assam Don Bosco University
Asha Hegde, Mangalore University
Ashutosh Tripathi, Publicis Sapient

Bagavathi C, Amrita Vishwa Vidyapeetham (Deemed University)
Bala Siva Sai Akhil Malepati, Independent
Benakeshkh Benakeshkh, National Institute of Technology Karnataka, India
Bharti Goel, F5 networks
Bhuvaneswari Sivagnanam, Central University of Tamil Nadu
Biniyam Lombe, Yale University
Brinda Gurusamy, Cisco
C. Oswald, NIT Tiruchirappalli
Chahat Raj, George Mason University
Chandrakanth Puligundla, Arizona State University
Chava Srinivasa Sai, Boston University
Chen-Yu Yen, ByteDance Inc.
Dao Sy Duy Minh, Ho Chi Minh City University of Science
Debajyoti Mazumder, IISER-Bhopal
Devi Sivakumar, University of California, Berkeley
Devpriya Dave, Bloomberg
Dhiman Goswami, George Mason University
Dipanjan Saha, Jadavpur University
Disne Kajanath, Eastern University of Sri Lanka
S. Delsi Robinsha, SRM Institute of Science and Technology
V.Sakthivel, Vellore Institute of Technology, India
Dss Lakshmi Kumari P, SRKR Engineering College
Farjana Yesmin, Boise State University
Fatima Uroosa, Instituto Politécnico Nacional
Fiona Victoria Stanley Jothiraj, Oregon State University
Garthigan Kumarasamy, University of Moratuwa, Sri Lanka
Gaurangi Sinha, Texas A&M University
Gersome Shimi, Madras Christian College
Ghanshyam Verma, University of Galway
H Mokrane, Observatoire de Paris
Habiba A, National Institute of Technology Puducherry, India
Hans Krupakar, Birla Institute of Technology and Science, Pilani
Hariharan R L, Vellore Institute of Technology
Hasan Murad, Chittagong University of Engineering and Technology
Hosahalli Lakshmaiah Shashirekha, Mangalore University
Jamaluddin Jamaluddin, Aligarh Muslim University, India
Jerin Mahibha C, Meenakshi Sundararajan Engineering College
Jobin Jose, Indian Institute of Information Technology Kottayam, India
Jubeerathan Thevakumar, University of Moratuwa
Judith Jeyafreeda Andrew, Université Paris Cité
Jyothish Lal G, Amrita Vishwa Vidyapeetham (Deemed University)
Kalaivani K S, Kongu Engineering College
Kamalakkannan Ravi, Harvard Medical School, Harvard University
Kamanasish Bhattacharjee, National Institute of Technology Agartala
Kanimozhi Selvi C S, Kongu Engineering College
Karthika Prabhu, Anna University
Karthika Subbaraj, Sri Sivasubramaniya Nadar Institutions
Keerthi Vasan A, RMK Engineering College, India
Kevlyn Kadamala, National University of Ireland, Galway
Kiran Pala, University of Eastern Finland
Koyel Ghosh, Central Institute of Technology Kokrajhar

Krishnakumari K, UNIVERSITY COLLEGE OF ENGINEERING, PANRUTI
Kuan Lu, Cornell University
Mamta Mamta, King's College London, University of London
Manan Mehta, University of Southern California
Manohar Sita Rama Madhurapantula, National Institute of Technical Teachers Training and Research
Manoj Balaji Jagadeeshan, Indian Institute of Technology Kharagpur, Dhirubhai Ambani Institute Of Information and Communication Technology
Martina Galletti, Sony Computer Science Laboratories-Paris
Md. Refaj Hossan, Chittagong University of Engineering and Technology
Megha Sundriyal, Max Planck Institute for Security and Privacy
Mehal Sakthi M S, Anna University, Thirunelveli
Menan Velayuthan, University of Moratuwa
Minoru Sasaki, Ibaraki University
Mithun Kumar S R, Birla Institute of Technology and Science
Mohammed Moshiul Hoque, Chittagong University of Engineering and Technology, Chittagong University of Engineering and Technology and Chittagong University of Engineering and Technology
Mohan Raj, Monash University, Malaysia
Mohanrasu S S, United Arab Emirates University
Momtazul Arefin Labib, Chittagong University of Engineering and Technology
Monorama Swain, Johannes Kepler Universität Linz
Msvpj Sathvik, University of Birmingham
Muhammad Arif
N.Nasurudeen Ahamed, College of Information Technology, United Arab Emirates University
Narthana Sivalingam, University of Moratuwa
Nazmus Sakib, Chittagong University of Engineering and Technology
Nida Hafeez, Instituto Politécnico Nacional, Mexico
Nida Saddaf Khan, University of Texas Health Center at Houson
Nilu R Salim, Sri Sivasubramaniya Nadar Institutions
Nitin Nikamanth Appiah Balaji, Hexion Inc.
Nurmyrat Amanmadov, University of Washington
Oleksii Cherkashyn, Blynk Technologies Inc.
Parthasarathi Jayaraman, SRM Institute Of Science & Technology (Deemed University)
Pengrui Han, Massachusetts Institute of Technology and California Institute of Technology
Prabalakshmi Arumugam, Boise State University
Prajna Prajna, Amity University
Pranav Gupta, Cisco
Prasanna Kumar Rangarajan, Amrita Vishwa Vidyapeetham
Prateek Jalan, Walmart
Pritam Deka, The Queen's University Belfast
R Tharaniya Sairaj, Indian Institute of Information Technology Kottayam, Indian Institute of Information Technology Kottayam
Radhe Shyam Salopanthula, Dhirubhai Ambani Institute Of Information and Communication Technology
Raghav Sharma, Workday
Rahul Seetharaman, LinkedIn
Rajat Dandekar, Vizuara
Rajat Patel, University of Maryland, Baltimore County
Rakesh Chandra Balabantaray, Indian Institute of Information Technology, Bhubaneswar, India
Rakesh Prakash, University of Colorado at Boulder

Ramakrishna Kolikipogu, Chaitanya Bharathi Institute of Technology
Ramesh Kannan R, Vellore Institute of Technology
Ratna Kandala, University of Kansas
Ratnajit Dhar, Chittagong University of Engineering and Technology
Ravi Teja Potla, NVIDIA
Rishabh Jain, eBay Inc.
Rizwana Kallooravi Thandil, Sullamussalam Science College Areekode
Rohan R
Sabita Langkam, SEEDS FINCAP PVT LTD
Sachin Sharma, Gran Sasso Science Institute
Sadiya Sayara Chowdhury Puspo, George Mason University
Sahil Rajesh Dhayalkar, Brain Corporation
Sai Kartheek Reddy Kasu, Colorado State University
Sai Koneru, Pennsylvania State University
Sampavi Jesuthas, University of Moratuwa, Sri Lanka
Sara Renjit, Indian Institute of Information Technology Kottayam, Indian Institute of Information Technology Kottayam
Sathurgini Uthayakumar, University of Moratuwa
Satya Sai Srinath Namburi Gnvv, GE HealthCare
Saurabh Aggarwal, Autodesk
Selam Abitte, Instituto Politécnico Nacional, Mexico
Seshu Babu Pulagara, National Institute of Technical Teachers Training and Research
Shaibal Saha, Oakland University
Shiti Chowdhury, Chittagong University of Engineering and Technology
Shouvik Kumar Guha, The West Bengal National University of Juridical Sciences
Shradha Sehgal, Department of Computer Science
Shravani Nag, Uber
Shubham Shubham, Birla Institute of Technology and Science, Hyderabad
Sidney Wong, University of Otago
Sikha Pentyala, University of Washington
Siranjeevi Rajamanickam, Government Polytechnic College, Trichy, India
Somsubhra De, Indian Institute of Technology, Roorkee
Soubraylu Sivakumar, SRM Institute of Science and Technology
Soumedhik Bharati, Sister Nivedita University
Sourabh Deoghare, Indian Institute of Technology Bombay, India
Sowbarnigaa Sowbarnigaa, Anna Univesity
Sowmya Vijayakumar, Technological University of the Shannon
Sreedath Panat, Vizuara AI Labs
Sreedeepa H S, Dhirubhai Ambani Institute Of Information and Communication Technology
Sreeja P. S, Vellore Institute of Technology
Sudhansu Bala Das, University of Galway, Ireland
Sudip Barik, Techno International New Town
Suresh Kumar Samarla, SRKR Engineering College
Surya Koppisetti, Reality Defender Inc
Sushovan Jena, Indian Institute of Technology Mandi, India
Swarup Ranjan Behera, ExxonMobil
Tasmiah Tahsin Mayeesha, University of North Texas
Tayyab Rehman, Università degli Studi dell'Aquila
Tewodros Achamaleh, Instituto Politécnico Nacional, Mexico
Tran Chi Nguyen
Trung-Kiet Huynh, Ho Chi Minh City University of Science, Vietnam

Tulika Bose, Vivoka
Uma Jothi, Amrita Vishwa Vidyapeetham (Deemed University)
Uthayasanker Thayasivam, University of Moratuwa
Vaibhava Lakshmi Ravideshik, GRAIL
Vani Kanjirangat, Dalle Molle Institute for Artificial Intelligence USI-SUPSI
Venkatesh Velugubantla, Meridian Cooperative
Vijayalakshmi P, Sri Sivasubramaniya Nadar Institutions
Vinay Babu Ulli, Oogwai Analytics
Vivek Pandit, Cadence Design Systems
Yazhmozhi V M, University of Dundee

Table of Contents

Abusive Content Detection in Telugu-English Code-Mixed Social Media Using Hybrid Transformer Architectures
Bojja Revanth Reddy and Sivaiah Bellamkonda 1

Beyond Benchmark Accuracy: Robustness Evaluation of Hinglish Sentiment Models
Chennuru Rahul and Kolawole Adebayo 6

Cascaded Modular or End-to-End? : An Investigation on Speech-to-Speech Translation Task for Dravidian Languages
Bhavana Nali and Abhik Jana 14

FLAICOL: Flip-Point-Led Augmentation for Imbalanced Code-Mixed Offensive Language Detection
Danish Mohammed and Vidhya Kamakshi 21

LIMP: Linguistically-Informed Multi-Strategy Prompting for Telugu Multi-Turn Dialogue Generation
Arjungopal Anilkumar, Suryansh Ram Menon, Divagar S and Premjith B 32

TamilMayangoliSpell: An Open-Source Neural Framework for Context-Sensitive Mayangoli Error Correction in Tamil
Yazhmozhi V M, Annalu Waller and Jacky Visser 42

TamilTok: Morphologically-Informed Tokenization for Tamil
Surendhar Muthukumar, Aaricia Herygers and Lisa Beinborn 52

Thiruppugazh-KG Dataset: A Manually Annotated Resource for Computational Analysis of Tamil Devotional Literature
Garthigan Kumarasamy, Jubeerathan Thevakumar, Sathurgini Uthayakumar, Disne Kajanath, Narthana Sivalingam and Uthayasanker Thayasivam 62

Findings in Tamil Dialect Speech Recognition and Classification
Bharathi B, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, Saranya S and Suhasini S 71

Findings of the Shared Task on Hope Speech Detection in Tulu
Thenmozhi Durairaj, Anusha M D Gowda, Raksha Adyanthaya, Rathnakara Shetty P, Parameshwar R Hegde, Mohammed Fadhel Aljunid, Prasanna Kumar Kumaresan and Bharathi Raja Chakravarthi 80

From Comments to Harm: A Findings Report on Abusive Tamil Text Targeting Women on Social Media Shared Task
Bhuvaneswari Sivagnanam, Kathiravan Pannerselvam, Jananayagan, Charmathi Rajkumar, Ramesh Kannan R, Ratnavel Rajalakshmi, Shunmuga Priya Muthusamy Chinnan, Saranya Rajiakodi and Bharathi Raja Chakravarthi 88

Overview of the Shared Task on Multilevel Political Meme Classification in Tamil and Malayalam
Saranya Rajiakodi, Shunmuga Priya Muthusamy Chinnan, Premjith B, Subalalitha CN, Rahul Ponnusamy, Anshid K A, Bhuvaneswari Sivagnanam and Bharathi Raja Chakravarthi 101

Shared Task on Depression Detection from Malayalam and Tamil Speech Data
Jyothish Lal G, Premjith B, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Thenmozhi Durairaj and Prasanna Kumar Kumaresan 114

Shared Task on Prompt Style Recovery for Large Language Models in Telugu
Premjith B, Jyothish Lal G, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Thenmozhi Durairaj, Ratnavel Rajalakshmi, Rahul Ponnusamy and Chinthala Bhuvanesh 124

TamilPoliSent 2026: A Shared Task report on Multiclass Political Sentiment Analysis in Tamil
Mani Vegupatti, Kishore Kumar Ponnusamy, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Thenmozhi Durairaj, Prasanna Kumar Kumaresan and Sathiyaraj Thangasamy 134

AbuseDetect_Alchemists@DravidianLangTech 2026: A Weighted Transformer Ensemble for Detecting Abusive Tamil Text Targeting Women
Meclin A Francis, Jyoti Kumari, Vinay Babu Ulli, Malavika Sreekumar and Joel Johnson . . . 144

AITamilDialect@DravidianLangTech 2026: Zero-Shot Whisper and Wav2Vec2 Embedding-Based Tamil Speech Recognition and Dialect Classification.
Varalakshmi K and Bharathi B 148

Azrael@DravidianLangTech 2026:Dialect-Sensitive Automatic Speech Recognition and Classification for Tamil
Janish Andrin J, Mohammed Sahil and Saranya S 153

ByteBreaker@DravidianLangTech 2026: XLM-RoBERTa Large with Sliding-Window Chunking and Top-K Mean Pooling for Writing Style Classification
Chava Srinivasa Sai, R Vinay Kumar, Jigeesha Sai Surapaneni and Chava Shanmukha Sai . . . 158

ByteBuilders@DravidianLangTech 2026: Transformer-Based Weighted Ensemble for Political Multiclass Sentiment Analysis of Tamil X (Twitter) Comments
Mitharshana T V, Shanthi S, Lavana V and Kaviya Varma R 163

cantnlp@DravidianLangTech 2026: organic domain adaptation improves multi-class hope speech detection in Tulu
Andrew Li and Sidney Wong 169

CHMOD_777@DravidianLangTech 2026: Context-Aware Fine-tuned MuRIL for Abusive Tamil Text Detection on Social Media
Arunaggiri Pandian Karunanidhi and Prabalakshmi Arumugam 176

CHMOD_777@DravidianLangTech 2026: LLM Augmented Transformer Fine-tuning for Tamil Political Sentiment Analysis
Arunaggiri Pandian Karunanidhi and Prabalakshmi Arumugam 181

CHMOD_777@DravidianLangTech 2026: Tamil-Adapted Whisper and MMS for Dialect Speech Recognition and Classification
Arunaggiri Pandian Karunanidhi and Prabalakshmi Arumugam 186

Cuet Yet Another Baseline@DravidianLangTech 2026: Shared Task on Prompt Recovery for LLM in Telugu
Rotna Dipika Debnath, Shahrin Afroz Hoque Ruhi, Ayesha Labiba, Arpita Mallik and Hasan Murad 191

CUET-2567@DravidianLangTech-ACL 2026: Multimodal Stance and Target Identification in Dravidian Political Memes
Arka Dutta, Anindya Majumder, Adnan Faisal and Hasan Murad 196

CUET_InferX@DravidianLangTech 2026: Shared Task on Dialect Based Speech Recognition and Classification in Tamil
Md. Ashraful Islam Semon, Jihadul Islam, Ratnajit Dhar and Hasan Murad 201

Cuet_Neural_Navigators@DravidianLangTech 2026: Depression Detection from Malayalam and Tamil Speech using Self-Supervised Acoustic Models
Shuva Dey, Abir Dey, Sha Newaz Mahmud and Hasan Murad 207

CUET_SYNTHETICA@DravidianLangTech 2026: Multi Architecture Transformer Ensemble for Detecting Abusive Tamil Text Targeting Women
Miftahul Jannat Rishta, Sumaiya Zaman, Shiti Chowdhury and Hasan Murad 212

CUET_SYNTHETICA@DravidianLangTech 2026: Multilingual Transformer Based Hope Speech Detection for Coarse and Fine-Grained Classification in Tulu
Sumaiya Zaman, Miftahul Jannat Rishta, Shiti Chowdhury and Hasan Murad 217

CYBERPUNK@DravidianLangTech 2026: Multimodal Political Meme Classification using CLIP and Logo Similarity
Shahad Abir 222

Dialectmind@DravidianLang Tech 2026: Zero-Shot Dialectal Tamil Automatic Speech Recognition Using a Large Pretrained Conformer Model
Gayathri.k and Bharathi B 227

DLRG@DravidianLangTech 2026: Dual-Purpose Whisper Adaptation for Tamil Dialect Identification and Dialectal Speech Recognition
Gulisetty Abhinav, Tanisha Nanda, Ramesh Kannan R and Ratnavel Rajalakshmi 232

DLRG@DravidianLangTech 2026: Explainable Transformer-Based Detection of Abusive Tamil Text Targeting Women on Social Media
Mirudhula Sankar and Ratnavel Rajalakshmi 237

DPR@DravidianLangTech 2026: Transformer-Based Abusive Content Detection for Tamil Text Targeting Women on Social Media
Diya Prakash, Praveen Kumar S, R Ranjith Kumar, Balasubramanian Palani, Jobin Jose and Siranjeevi Rajamanickam 242

Dravid-Tech-Builders@DravidianLangTech 2026: A Comparative Study of Classical and Deep Learning Approaches for Tamil Dialect Classification and Speech Recognition
Naveen A, Karthiyayini P and Kalaivani K S 248

ERROR_500@DravidianLangTech2026: Automatic Prompt Style Classification in Telugu Using Transformer-Based Language Models
Mahashweta Manjari Barua, Tasnia Khanam, Nuzha Saifa Rahmat, Shiti Chowdhury and Hasan Murad 253

HNK@DravidianLangTech 2026: Investigating Grapheme-Level Normalization for Abusive Tamil Text Classification
Hanish Vigneshwar R, Nahul Alaguraj, Karthikeyan Manimaran and Ratnavel Rajalakshmi . . 258

Hope_Speech_Alchemists@DravidianLangTech 2026: TF-IDF SVM and XLM-RoBERTa with Focal Loss for Hope Speech Detection in Tulu
Joel Johnson, Meclin A Francis, Jyoti Kumari, Malavika Sreekumar and Vinay Babu Ulli . . . 263

IIITK_SpeechScape@DravidianLangTech 2026: Dialect based speech recognition and classification using Speech Foundation Models and Deep Learning Techniques
G Srishtik Sekar, Harissh Ragav Dhamodaran, Kishore Shankar S, Balasubramanian Palani and R Tharaniya Sairaj 268

IndiLangTech@DravidianLangTech 2026: Hierarchical Modeling for Multi-Level Political Meme Classification
Saurabh Kumar, Vivekananda G, Ranbir Singh Sanasam and Sukumar Nandi 273

JerinWarriors@DravidianLangTech 2026: A Two-Stream Cross-Attention Approach for Prompt Recovery in Telugu
Savith A, Wordson Robert, Jerin Mahibha C and Shrey Patnaik 279

KEC'S CODE CRAFTERS@DravidianLangTech 2026: Abusive Tamil Text Detection Targeting Women on Social Media
Nivetha, Nethrasri S and Malliga Subramanian 284

Lannisters@DravidianLangTech 2026: A Comparative and Ablation Study of Multilingual Transformers for Gender-Targeted Abuse Detection in Tamil Social Media Platforms
Kalaivani K S, Jaisanth K and Nandhini B 289

Mano_sub@DravidianLangTech 2026: Article-Aware Batching and Discriminative Fine-Tuning of MuRIL for Telugu Prompt-Style Classification
Manohar Sita Rama Madhurapantula and Seshu Babu Pulagara 294

MedHastra@DravidianLangTech 2026: Piecewise Style Classification for Telugu Prompt Recovery Using XLM-RoBERTa
Shruti Chandrasekar, Vedajanaani R S and Vijayalakshmi P 301

MUCS@Dravidianlangtech@ACL2026: Hope Speech Detection in Code-Mixed Tulu Language Using Multiple Features
Hosahalli Lakshmaiah Shashirekha and Rachana A 306

NITC-HSR@DravidianLangTech 2026: Ensembling Multilingual Transformer Models for Detecting Abusive Tamil Text Targeting Women on Social Media
Rameez Mohammed A and S D Madhu Kumar 316

PhucNguyen@DravidianLangTech 2026: Political Multiclass Sentiment Analysis with XLM-RoBERTa and Low-Rank Adaptation
Dinh Khac Phuc Nguyen and Thìn Đặng Văn 321

PolyTicsTamil_Alchemists@DravidianLangTech@ACL 2026: An Augmentation-Driven Focal Ensemble Model for Political Sentiment Analysis in Tamil
Jyoti Kumari, Meclin A Francis, Vinay Babu Ulli, Malavika Sreekumar and Joel Johnson . . . 326

PrimeLine@DravidianLangTech 2026: Abusive Tamil Comment Detection Using MuRIL
Rithikaa V, S.Sumathi, Nithya Varshini C N R and Sanjay Krishnan K 331

PrimeLine@DravidianLangTech 2026: Hope Speech Detection in Tulu Using XLM-RoBERTa for Coarse and Fine-Grained Classification
Rithikaa V, S.Sumathi, Sanjay Krishnan K and Nithya Varshini C N R 336

RMS@DravidianLangTech 2026: Multimodal Gated Fusion for Hierarchical Tamil Political Meme Classification
Md. Ajwad Hossain 341

Semantica@DravidianLangTech 2026: Vision-Language Models for Hierarchical Political Meme Classification in Tamil and Malayalam
Junain Uddin, Rahul Datta, Taha Ibne Abdullah and Hasan Murad 348

SERENE@DravidianLangTech 2026: Multimodal Approaches for Depression Detection in Dravidian Speech: Acoustic, Spectrogram, and Transformer-Based Models
TT Pranesh, K.K.Thamizhmathi, S Vigneshwaran and Bharathi B 354

SJM_MINDS@DravidianLangTech@ACL2026: Machine Learning Approaches for Hope Speech Detection in Code-Mixed Tulu
Hosahalli Lakshmaiah Shashirekha, Manjula and Jayashree Krishna 359

SSN_HopeNetters@DravidianLangTech 2026: Multi-Level Hope Speech Detection using XLM-RoBERTa
Moogambigai A, Bharathi B, Nikhil Karthik S, Pandiarajan D and Nandhika Saravanan 366

Still Loading@DravidianLangTech 2026: Telugu Prompt-Style Recovery using Multilingual Transformers
Samonwita Sarker, Isnat Mehrin Sami, Priyontee Mojumder, Arpita Mallik and Hasan Murad371

SUPERNOVA@DravidianLangTech 2026: Transformer and Ensemble Approaches for Abusive Tamil Text Detection Targeting Women
Kiruthika K, Roahiyaa T and Premjith B 376

SYNAPSE@DravidianLangTech 2026: Multi-Level Political Meme Classification for Tamil and Malayalam
Suriya KP, Durai Singh K, Gnanasabesan G, Ganesh Sundhar S, Hari Krishnan N and Jyothish Lal G 381

TamilEcho_Political@DravidianLangTech 2026: Hybrid XLM-RoBERTa with Sarcasm-Aware Feature Fusion for Political Multiclass Sentiment Analysis in Tamil X
Kanimozhi Selvi C S, Inigashree N S, Kavinraj J and Moneissh A G 387

TAMILGOODBADTXT@DravidianLangTech 2026:A Multilingual Transformer-Based Approach for Abusive Language Identification in Tamil Social Media
Varalakshmi K and Bharathi B 392

TamilVoiceLab@DravidianLangTech 2026: Investigating Whisper Tamil Large-v2 for Dialectal Tamil Speech Recognition
S.b.priya and Bharathi B 397

Team Oryu@DravidianLangTech 2026: A Multilingual Transformer Approach for Hope Speech Detection in Code-Mixed Tulu
Joyeta Barua Moni, Noore Tamanna Orny, Md. Abtahee Kabir and Hasan Murad 403

Team_One@DravidianLangTech 2026: A Gated Multimodal Architecture for Multi-Level Stance and Target Detection in Malayalam Political Memes
Nimisha M Iyer, Ashmi S N, Balasubramanian Palani, Jobin Jose and Siranjeevi Rajamanickam 409

Trailblazer@DravidianLangTech 2026: A Comparative Study of TF-IDF SVM and XLM-RoBERTa for Political Multiclass Text Classification.
Anuradha C, Anbuaruvi R and Shanthi Murugan 414

TriVector@DravidianLangTech 2026: Abusive Tamil Text Detection on Social Media Using Lexicon-Augmented Transformers
Oarisa Rebayet, Tahmima Hoque Eid, Fawzia Tabassum and Hasan Murad 420

TriVector@DravidianLangTech 2026: Depression Detection from Tamil and Malayalam Speech with Speaker-Independent Evaluation using MFCC and Wav2Vec2
Tahmima Hoque Eid, Fawzia Tabassum, Oarisa Rebayet and Hasan Murad 429

VITECH@DravidianLangTech2026: Prompting and LoRA Adaptation for Tamil Abusive Language Detection - A Comparative Study of Open LLMs
Triambiga Krubhakaran, Senthil Kumar B, Kaviya Nagarajan and Balaji N 436

Wave2Word@DravidianLangTech 2026: WhisTam: A unified framework for dialect based Tamil speech recognition and classification
Ruwad Naswan and Shadab Tanjeed Ahmad 442

Wise@DravidianLangTech 2026: Dialect-Aware Tamil Speech Classification and Recognition via Cross-Pipeline Embedding Transfer
Ganesh Sundhar S, Hari Krishnan N, Gnanasabesan G, Suriya KP and Jyothish Lal G 447

Program

Friday, July 3, 2026

09:00 - 09:15 *Opening Remarks (All times listed in Pacific Time - San Diego, California, United States)*

09:15 - 10:30 *Oral Session 1*

09:15 - 09:30 *Abusive Content Detection in Telugu-English Code-Mixed Social Media Using Hybrid Transformer Architectures*
Bojja Revanth Reddy and Sivaiah Bellamkonda

09:30 - 09:45 *Beyond Benchmark Accuracy: Robustness Evaluation of Hinglish Sentiment Models*
Chennuru Rahul and Kolawole Adebayo

09:45 - 10:00 *Cascaded Modular or End-to-End? : An Investigation on Speech-to-Speech Translation Task for Dravidian Languages*
Bhavana Nali and Abhik Jana

10:00 - 10:15 *FLAICOL: Flip-Point-Led Augmentation for Imbalanced Code-Mixed Offensive Language Detection*
Danish Mohammed and Vidhya Kamakshi

10:30 - 11:00 *Coffee Break*

11:00 - 12:00 *Oral Session 2*

11:00 - 11:15 *LIMP: Linguistically-Informed Multi-Strategy Prompting for Telugu Multi-Turn Dialogue Generation*
Arjungopal Anilkumar, Suryansh Ram Menon, Divagar S and Premjith B

11:15 - 11:30 *TamilMayangoliSpell: An Open-Source Neural Framework for Context-Sensitive Mayangoli Error Correction in Tamil*
Yazhmozhi V M, Annalu Waller and Jacky Visser

11:30 - 11:45 *TamilTok: Morphologically-Informed Tokenization for Tamil*
Surendhar Muthukumar, Aaricia Herygers and Lisa Beinborn

11:45 - 12:00 *Thiruppugazh-KG Dataset: A Manually Annotated Resource for Computational Analysis of Tamil Devotional Literature*
Garthigan Kumarasamy, Jubeerathan Thevakumar, Sathurgini Uthayakumar, Disne Kajanath, Narthana Sivalingam and Uthayasanker Thayasivam

Friday, July 3, 2026 (continued)

12:00 - 13:30 *Lunch Break*

13:30 - 15:30 *Oral Session 3*

13:30 - 13:45 *Findings in Tamil Dialect Speech Recognition and Classification*
Bharathi B, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, Saranya S and Suhasini S

13:45 - 14:00 *Findings of the Shared Task on Hope Speech Detection in Tulu*
Thenmozhi Durairaj, Anusha M D Gowda, Raksha Adyanthaya, Rathnakara Shetty P, Parameshwar R Hegde, Mohammed Fadhel Aljunid, Prasanna Kumar Kumaresan and Bharathi Raja Chakravarthi

14:00 - 14:15 *From Comments to Harm: A Findings Report on Abusive Tamil Text Targeting Women on Social Media Shared Task*
Bhuvaneswari Sivagnanam, Kathiravan Pannerselvam, Jananayagan, Charmathi Rajkumar, Ramesh Kannan R, Ratnavel Rajalakshmi, Shunmuga Priya Muthusamy Chinnan, Saranya Rajiakodi and Bharathi Raja Chakravarthi

14:15 - 14:30 *Overview of the Shared Task on Multilevel Political Meme Classification in Tamil and Malayalam*
Saranya Rajiakodi, Shunmuga Priya Muthusamy Chinnan, Premjith B, Subalalitha CN, Rahul Ponnusamy, Anshid K A, Bhuvaneswari Sivagnanam and Bharathi Raja Chakravarthi

14:30 - 14:45 *Shared Task on Depression Detection from Malayalam and Tamil Speech Data*
Jyothish Lal G, Premjith B, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Thenmozhi Durairaj and Prasanna Kumar Kumaresan

14:45 - 15:00 *Shared Task on Prompt Style Recovery for Large Language Models in Telugu*
Premjith B, Jyothish Lal G, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Thenmozhi Durairaj, Ratnavel Rajalakshmi, Rahul Ponnusamy and Chinthala Bhuvanesh

15:00 - 15:15 *TamilPoliSent 2026: A Shared Task report on Multiclass Political Sentiment Analysis in Tamil*
Mani Vegupatti, Kishore Kumar Ponnusamy, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Thenmozhi Durairaj, Prasanna Kumar Kumaresan and Sathiyaraj Thangasamy

15:15 - 15:30 *AbuseDetect_Alchemists@DravidianLangTech 2026: A Weighted Transformer Ensemble for Detecting Abusive Tamil Text Targeting Women*
Meclin A Francis, Jyoti Kumari, Vinay Babu Ulli, Malavika Sreekumar and Joel Johnson

15:30 - 16:00 *Coffee Break*

16:00 - 17:00 *Poster Session*

Friday, July 3, 2026 (continued)

16:00 - 16:15 *AITamilDialect@DravidianLangTech 2026: Zero-Shot Whisper and Wav2Vec2 Embedding-Based Tamil Speech Recognition and Dialect Classification.*
Varalakshmi K and Bharathi B

16:15 - 16:30 *Azrael@DravidianLangTech 2026:Dialect-Sensitive Automatic Speech Recognition and Classification for Tamil*
Janish Andrin J, Mohammed Sahil and Saranya S

16:30 - 16:45 *ByteBreaker@DravidianLangTech 2026: XLM-RoBERTa Large with Sliding-Window Chunking and Top-K Mean Pooling for Writing Style Classification*
Chava Srinivasa Sai, R Vinay Kumar, Jigeesha Sai Surapaneni and Chava Shanmukha Sai

16:45 - 17:00 *ByteBuilders@DravidianLangTech 2026: Transformer-Based Weighted Ensemble for Political Multiclass Sentiment Analysis of Tamil X (Twitter) Comments*
Mitharshana T V, Shanthi S, Lavana V and Kaviya Varma R

17:00 - 17:30 *Poster Session*

17:00 - 17:30 *cantnlp@DravidianLangTech 2026: organic domain adaptation improves multiclass hope speech detection in Tulu*
Andrew Li and Sidney Wong

17:00 - 17:30 *CHMOD_777@DravidianLangTech 2026: Context-Aware Fine-tuned MuRIL for Abusive Tamil Text Detection on Social Media*
Arunaggiri Pandian Karunanidhi and Prabalakshmi Arumugam

17:00 - 17:30 *CHMOD_777@DravidianLangTech 2026: LLM Augmented Transformer Fine-tuning for Tamil Political Sentiment Analysis*
Arunaggiri Pandian Karunanidhi and Prabalakshmi Arumugam

17:00 - 17:30 *CHMOD_777@DravidianLangTech 2026: Tamil-Adapted Whisper and MMS for Dialect Speech Recognition and Classification*
Arunaggiri Pandian Karunanidhi and Prabalakshmi Arumugam

17:00 - 17:30 *Cuet Yet Another Baseline@DravidianLangTech 2026: Shared Task on Prompt Recovery for LLM in Telugu*
Rotna Dipika Debnath, Shahrin Afroz Hoque Ruhi, Ayesha Labiba, Arpita Mallik and Hasan Murad

17:00 - 17:30 *CUET-2567@DravidianLangTech-ACL 2026: Multimodal Stance and Target Identification in Dravidian Political Memes*
Arka Dutta, Anindya Majumder, Adnan Faisal and Hasan Murad

Friday, July 3, 2026 (continued)

17:00 - 17:30 *CUET_InferX@DravidianLangTech 2026: Shared Task on Dialect Based Speech Recognition and Classification in Tamil*
Md. Ashraful Islam Semon, Jihadul Islam, Ratnajit Dhar and Hasan Murad

17:00 - 17:30 *Cuet_Neural_Navigators@DravidianLangTech 2026: Depression Detection from Malayalam and Tamil Speech using Self-Supervised Acoustic Models*
Shuva Dey, Abir Dey, Sha Newaz Mahmud and Hasan Murad

17:00 - 17:30 *CUET_SYNTHETICA@DravidianLangTech 2026: Multi Architecture Transformer Ensemble for Detecting Abusive Tamil Text Targeting Women*
Miftahul Jannat Rishta, Sumaiya Zaman, Shiti Chowdhury and Hasan Murad

17:00 - 17:30 *CUET_SYNTHETICA@DravidianLangTech 2026: Multilingual Transformer Based Hope Speech Detection for Coarse and Fine-Grained Classification in Tulu*
Sumaiya Zaman, Miftahul Jannat Rishta, Shiti Chowdhury and Hasan Murad

17:00 - 17:30 *CYBERPUNK@DravidianLangTech 2026: Multimodal Political Meme Classification using CLIP and Logo Similarity*
Shahad Abir

17:00 - 17:30 *Dialectmind@DravidianLang Tech 2026: Zero-Shot Dialectal Tamil Automatic Speech Recognition Using a Large Pretrained Conformer Model*
Gayathri.k and Bharathi B

17:00 - 17:30 *DLRG@DravidianLangTech 2026: Dual-Purpose Whisper Adaptation for Tamil Dialect Identification and Dialectal Speech Recognition*
Gulisetty Abhinav, Tanisha Nanda, Ramesh Kannan R and Ratnavel Rajalakshmi

17:00 - 17:30 *DLRG@DravidianLangTech 2026: Explainable Transformer-Based Detection of Abusive Tamil Text Targeting Women on Social Media*
Mirudhula Sankar and Ratnavel Rajalakshmi

17:00 - 17:30 *DPR@DravidianLangTech 2026: Transformer-Based Abusive Content Detection for Tamil Text Targeting Women on Social Media*
Diya Prakash, Praveen Kumar S, R Ranjith Kumar, Balasubramanian Palani, Jobin Jose and Siranjeevi Rajamanickam

17:00 - 17:30 *Dravid-Tech-Builders@DravidianLangTech 2026: A Comparative Study of Classical and Deep Learning Approaches for Tamil Dialect Classification and Speech Recognition*
Naveen A, Karthiyayini P and Kalaivani K S

17:00 - 17:30 *ERROR_500@DravidianLangTech2026: Automatic Prompt Style Classification in Telugu Using Transformer-Based Language Models*
Mahashweta Manjari Barua, Tasnia Khanam, Nuzha Saifa Rahmat, Shiti Chowdhury and Hasan Murad

Friday, July 3, 2026 (continued)

17:00 - 17:30 *HNK@DravidianLangTech 2026: Investigating Grapheme-Level Normalization for Abusive Tamil Text Classification*
Hanish Vigneshwar R, Nahul Alaguraj, Karthikeyan Manimaran and Ratnavel Rajalakshmi

17:00 - 17:30 *Hope_Speech_Alchemists@DravidianLangTech 2026: TF-IDF SVM and XLM-RoBERTa with Focal Loss for Hope Speech Detection in Tulu*
Joel Johnson, Meclin A Francis, Jyoti Kumari, Malavika Sreekumar and Vinay Babu Ulli

17:00 - 17:30 *IIITK_SpeechScape@DravidianLangTech 2026: Dialect based speech recognition and classification using Speech Foundation Models and Deep Learning Techniques*
G Srishtik Sekar, Harissh Ragav Dhamodaran, Kishore Shankar S, Balasubramanian Palani and R Tharaniya Sairaj

17:00 - 17:30 *IndiLangTech@DravidianLangTech 2026: Hierarchical Modeling for Multi-Level Political Meme Classification*
Saurabh Kumar, Vivekananda G, Ranbir Singh Sanasam and Sukumar Nandi

17:00 - 17:30 *JerinWarriors@DravidianLangTech 2026: A Two-Stream Cross-Attention Approach for Prompt Recovery in Telugu*
Savith A, Wordson Robert, Jerin Mahibha C and Shrey Patnaik

17:00 - 17:30 *KEC'S CODE CRAFTERS@DravidianLangTech 2026: Abusive Tamil Text Detection Targeting Women on Social Media*
Nivetha, Nethrasri S and Malliga Subramanian

17:00 - 17:30 *Lannisters@DravidianLangTech 2026: A Comparative and Ablation Study of Multilingual Transformers for Gender-Targeted Abuse Detection in Tamil Social Media Platforms*
Kalaivani K S, Jaisanth K and Nandhini B

17:00 - 17:30 *Mano_sub@DravidianLangTech 2026: Article-Aware Batching and Discriminative Fine-Tuning of MuRIL for Telugu Prompt-Style Classification*
Manohar Sita Rama Madhurapantula and Seshu Babu Pulagara

17:00 - 17:30 *MedHastra@DravidianLangTech 2026: Piecewise Style Classification for Telugu Prompt Recovery Using XLM-RoBERTa*
Shruti Chandrasekar, Vedajanaani R S and Vijayalakshmi P

17:00 - 17:30 *MUCS@Dravidianlangtech@ACL2026: Hope Speech Detection in Code-Mixed Tulu Language Using Multiple Features*
Hosahalli Lakshmaiah Shashirekha and Rachana A

17:00 - 17:30 *NITC-HSR@DravidianLangTech 2026: Ensembling Multilingual Transformer Models for Detecting Abusive Tamil Text Targeting Women on Social Media*
Rameez Mohammed A and S D Madhu Kumar

Friday, July 3, 2026 (continued)

17:00 - 17:30 *PhucNguyen@DravidianLangTech 2026: Political Multiclass Sentiment Analysis with XLM-RoBERTa and Low-Rank Adaptation*
Dinh Khac Phuc Nguyen and Thìn Đặng Văn

17:00 - 17:30 *PolyTicsTamil_Alchemists@DravidianLangTech@ACL 2026: An Augmentation-Driven Focal Ensemble Model for Political Sentiment Analysis in Tamil*
Jyoti Kumari, Meclin A Francis, Vinay Babu Ulli, Malavika Sreekumar and Joel Johnson

17:00 - 17:30 *PrimeLine@DravidianLangTech 2026: Abusive Tamil Comment Detection Using MuRIL*
Rithikaa V, S.Sumathi, Nithya Varshini C N R and Sanjay Krishnan K

17:00 - 17:30 *PrimeLine@DravidianLangTech 2026: Hope Speech Detection in Tulu Using XLM-RoBERTa for Coarse and Fine-Grained Classification*
Rithikaa V, S.Sumathi, Sanjay Krishnan K and Nithya Varshini C N R

17:00 - 17:30 *RMS@DravidianLangTech 2026: Multimodal Gated Fusion for Hierarchical Tamil Political Meme Classification*
Md. Ajwad Hossain

17:00 - 17:30 *Semantica@DravidianLangTech 2026: Vision-Language Models for Hierarchical Political Meme Classification in Tamil and Malayalam*
Junain Uddin, Rahul Datta, Taha Ibne Abdullah and Hasan Murad

17:00 - 17:30 *SERENE@DravidianLangTech 2026: Multimodal Approaches for Depression Detection in Dravidian Speech: Acoustic, Spectrogram, and Transformer-Based Models*
TT Pranesh, K.K.Thamizhmathi, S Vigneshwaran and Bharathi B

17:00 - 17:30 *SJM_MINDS@DravidianLangTech@ACL2026: Machine Learning Approaches for Hope Speech Detection in Code-Mixed Tulu*
Hosahalli Lakshmaiah Shashirekha, Manjula and Jayashree Krishna

17:00 - 17:30 *SSN_HopeNetters@DravidianLangTech 2026: Multi-Level Hope Speech Detection using XLM-RoBERTa*
Moogambigai A, Bharathi B, Nikhil Karthik S, Pandiarajan D and Nandhika Saravanan

17:00 - 17:30 *Still Loading@DravidianLangTech 2026: Telugu Prompt-Style Recovery using Multilingual Transformers*
Samonwita Sarker, Isnat Mehrin Sami, Priyontee Mojumder, Arpita Mallik and Hasan Murad

17:00 - 17:30 *SUPERNOVA@DravidianLangTech 2026: Transformer and Ensemble Approaches for Abusive Tamil Text Detection Targeting Women*
Kiruthika K, Roahiyaa T and Premjith B

Friday, July 3, 2026 (continued)

17:00 - 17:30 *SYNAPSE@DravidianLangTech 2026: Multi-Level Political Meme Classification for Tamil and Malayalam*
Suriya KP, Durai Singh K, Gnanasabesan G, Ganesh Sundhar S, Hari Krishnan N and Jyothish Lal G

17:00 - 17:30 *TamilEcho_Political@DravidianLangTech 2026: Hybrid XLM-RoBERTa with Sarcasm-Aware Feature Fusion for Political Multiclass Sentiment Analysis in Tamil X*
Kanimozhi Selvi C S, Inigashree N S, Kavinraj J and Moneissh A G

17:00 - 17:30 *TAMILGOODBADTXT@DravidianLangTech 2026:A Multilingual Transformer-Based Approach for Abusive Language Identification in Tamil Social Media*
Varalakshmi K and Bharathi B

17:00 - 17:30 *TamilVoiceLab@DravidianLangTech 2026: Investigating Whisper Tamil Large-v2 for Dialectal Tamil Speech Recognition*
S.b.priya and Bharathi B

17:00 - 17:30 *Team Oryu@DravidianLangTech 2026: A Multilingual Transformer Approach for Hope Speech Detection in Code-Mixed Tulu*
Joyeta Barua Moni, Noore Tamanna Orny, Md. Abtahee Kabir and Hasan Murad

17:00 - 17:30 *Team_One@DravidianLangTech 2026: A Gated Multimodal Architecture for Multi-Level Stance and Target Detection in Malayalam Political Memes*
Nimisha M Iyer, Ashmi S N, Balasubramanian Palani, Jobin Jose and Siranjeevi Rajamanickam

17:00 - 17:30 *Trailblazer@DravidianLangTech 2026: A Comparative Study of TF-IDF SVM and XLM-RoBERTa for Political Multiclass Text Classification.*
Anuradha C, Anbuaruvi R and Shanthi Murugan

17:00 - 17:30 *TriVector@DravidianLangTech 2026: Abusive Tamil Text Detection on Social Media Using Lexicon-Augmented Transformers*
Oarisa Rebayet, Tahmima Hoque Eid, Fawzia Tabassum and Hasan Murad

17:00 - 17:30 *TriVector@DravidianLangTech 2026: Depression Detection from Tamil and Malayalam Speech with Speaker-Independent Evaluation using MFCC and Wav2Vec2*
Tahmima Hoque Eid, Fawzia Tabassum, Oarisa Rebayet and Hasan Murad

17:00 - 17:30 *VITECH@DravidianLangTech2026: Prompting and LoRA Adaptation for Tamil Abusive Language Detection - A Comparative Study of Open LLMs*
Triambiga Krubhakaran, Senthil Kumar B, Kaviya Nagarajan and Balaji N

17:00 - 17:30 *Wave2Word@DravidianLangTech 2026: WhisTam: A unified framework for dialect based Tamil speech recognition and classification*
Ruwad Naswan and Shadab Tanjeed Ahmad

Friday, July 3, 2026 (continued)

17:00 - 17:30 *Wise@DravidianLangTech 2026: Dialect-Aware Tamil Speech Classification and Recognition via Cross-Pipeline Embedding Transfer*
Ganesh Sundhar S, Hari Krishnan N, Gnanasabesan G, Suriya KP and Jyothish Lal G

17:30 - 17:45 *Closing Remarks*

Abusive Content Detection in Telugu-English Code-Mixed Social Media Using Hybrid Transformer Architectures

Bojja Revanth Reddy, Sivaiah Bellamkonda

Abstract

The rapid growth of social media has led to a rise in abusive and offensive content, especially in code-mixed languages like Telugu-English. Detecting such content is challenging because of transliteration, spelling variations, informal writing styles, and frequent switching between languages. This paper presents a hybrid deep learning model for abusive content detection in Telugu-English code-mixed comments. A pretrained transformer model, DeBERTa, is used to generate contextual embeddings, which are further processed through parallel Multi-Channel CNN and BiLSTM networks for feature extraction. The CNN captures important local patterns and phrases, while the BiLSTM learns sequential and contextual information from the text. The combined features are then used for final classification. The proposed model effectively handles the linguistic complexity of Telugu-English code-mixed social media data and improves abusive content detection performance.

1 Introduction

Social media platforms have transformed global communication by enabling real-time interaction and large-scale content sharing. While these platforms facilitate engagement and information exchange, they also enable the spread of abusive language, abusive speech, cyberbullying, and misinformation. Due to the massive volume of user-generated content produced daily, manual moderation is infeasible, making automated detection systems essential.

Artificial Intelligence (AI) and Natural Language Processing (NLP) techniques are widely adopted for automated abusive content detection. However, detecting abusive language in Telugu-English code-mixed social media text presents additional challenges. Code-mixing refers to the blending of two languages within a single sentence or discourse. In Telugu-English contexts, users frequently write Telugu words using the Roman script, resulting in inconsistent transliterations and spelling variations.

For example, a single Telugu word may appear in multiple Romanized forms depending on user preference. Informal grammar, slang, repeated characters, and creative spellings further complicate tokenization and feature extraction. Additionally, contextual ambiguity and sarcasm make it difficult to distinguish between genuine and abusive intent.

Traditional NLP systems trained on monolingual corpora struggle to generalize in such multilingual and noisy environments. Therefore, robust preprocessing techniques, effective feature representations, and contextual deep learning models are necessary for reliable abusive content detection in Telugu-English code-mixed data.

2 Literature Survey

Recent research in abusive content detection has evolved from keyword-based filtering to machine learning and deep learning approaches. Traditional supervised models such as Naïve Bayes, Support Vector Machines (SVM), and Logistic Regression rely on feature engineering techniques like n-grams and TF-IDF representations. These methods provide strong baseline performance but often fail to capture contextual nuances in complex language settings.

Deep learning models, including CNNs and LSTMs, improve performance by learning hierarchical representations of text. More recently, transformer-based architectures such as BERT, XLM-RoBERTa, and IndicBERT have demonstrated superior performance in multilingual and low-resource scenarios. These models leverage subword tokenization and self-attention mechanisms to capture contextual dependencies and handle spelling variations.

However, Telugu-English code-mixed abusive

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 1–5
July 3, 2026 ©2026 Association for Computational Linguistics

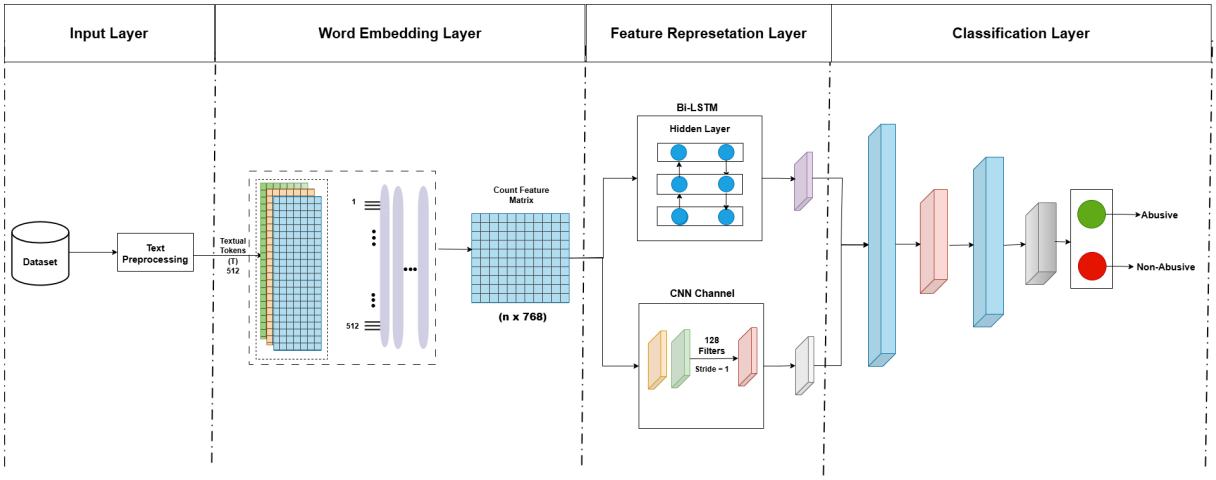


Figure 1: Hybrid transformer–CNN–BiLSTM architecture.

detection remains underexplored due to limited annotated datasets, transliteration variability, and morphological complexity of Telugu. This gap highlights the need for domain-specific preprocessing and fine-tuned multilingual transformer models tailored to code-mixed environments.

3 Methodology

3.1 Preprocessing

Preprocessing was performed to ensure textual consistency and quality before model training. Telugu-English code-mixed social media text often contains noise, informal expressions, spelling variations, and inconsistent transliterations. Therefore, multiple normalization steps were applied. Missing values, URLs, emojis, and extra whitespace were removed during data cleaning. Special symbols and unnecessary punctuation marks were eliminated to reduce noise. Common Telugu-English stopwords were filtered out to minimize redundancy and enhance discriminative learning.

To address lexical inconsistency, variant spellings and abusive term synonyms were standardized to maintain uniform representation across samples. Furthermore, after data augmentation, native Telugu script was converted back into Romanized Telugu script to ensure consistent feature extraction and compatibility with the model tokenizer. These preprocessing steps ensured uniform textual representation and improved interpretability and downstream classification performance.

3.2 Dataset Augmentation

To improve linguistic diversity and model robustness, data augmentation techniques were applied. The final augmented dataset consists of 49,910 comments, providing broader coverage of both abusive and non-abusive expressions. Augmentation included synonym substitution, controlled word insertion and deletion, and generation of spelling and dialectal variants commonly observed in Telugu-English code-mixed social media text. These techniques simulate natural user behavior and enhance the model's ability to generalize to unseen variations of abusive language.

3.3 Model Architecture

The proposed model integrates transformer-based contextual embeddings with convolutional and recurrent neural networks to capture both local and global linguistic patterns. A multilingual transformer encoder first converts input text into contextualized embeddings, capturing semantic relationships and multilingual characteristics of code-mixed data.

The contextual embeddings are then passed through a Convolutional Neural Network (CNN) layer to extract local n-gram and phrase-level patterns using convolution and max-pooling operations. In parallel, a Bidirectional Long Short-Term Memory (BiLSTM) network processes the embeddings to capture long-range sequential dependencies by analyzing text in both forward and backward directions.

The outputs from the CNN and BiLSTM layers are concatenated to form a rich fused representation that combines local lexical features and global contextual information. This fused feature vector is passed through a fully connected dense layer with a Sigmoid activation function to perform binary clas-

sification and predict whether the input comment is abusive or non-abusive.

This hybrid architecture enables effective modeling of contextual semantics, phrase-level patterns, and sequential dependencies, making it well-suited for abusive content detection in Telugu-English code-mixed social media text.

4 Experiment

This section presents the experimental configuration, dataset details, and evaluation metrics used to assess the performance of the proposed abusive content detection model for Telugu-English code-mixed social media text.

4.1 Experimental Setup

All experiments were conducted using Jupyter Notebook, which provides an integrated environment for coding, visualization, and model evaluation. The models were implemented using the TensorFlow framework with Keras as the high-level API for constructing neural network architectures.

The proposed hybrid architecture integrates a multilingual transformer encoder with CNN and BiLSTM layers. Transformer-based models such as XLM-RoBERTa and IndicBERT were used to generate contextual embeddings. The dataset was divided into 75% training data and 25% testing data to ensure reliable evaluation of generalization performance.

The models were trained using binary cross-entropy loss with the Adam optimizer. Early stopping was employed to prevent overfitting and ensure stable convergence.

4.2 Dataset

The dataset consists of Telugu-English code-mixed social media comments annotated for abusive (abusive) and non-abusive (non-abusive) categories. The data contains transliterated Telugu text written in Roman script along with English words, reflecting realistic informal online communication.

After preprocessing and augmentation, the final dataset contains a total of **49,910 comments**. The class distribution of the final augmented dataset is shown in Table 1.

The augmented dataset provides improved linguistic diversity and balanced representation of abusive and non-abusive expressions, enabling better generalization of the classification model.

Table 1: Final Augmented Dataset Distribution

Class	Label	Number of Samples
Non-abusive	0	23,342
Abusive	1	26,568
Total		**49,910**

4.3 Performance Metrics

The models were evaluated using standard classification metrics, including accuracy, precision, recall, and F1 score.

1. **Accuracy**:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

2. **Precision**:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

3. **Recall**:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

4. **F1 Score**:

$$F1 = 2 \cdot \left(\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \right) \tag{4}$$

Here, **TP** denotes True Positives, **TN** denotes True Negatives, **FP** denotes False Positives, and **FN** denotes False Negatives. Since abusive content detection may involve class imbalance, macro-averaged F1 score is emphasized to ensure fair performance evaluation across both classes.

5 Results

This section presents the quantitative evaluation and analytical insights of the proposed hybrid architecture for abusive content detection in Telugu-English code-mixed text.

5.1 Quantitative Performance

To evaluate the impact of different embedding backbones combined with additional feature extraction layers, we compare transformer-based contextual encoders integrated with parallel CNN–BiLSTM modules. The CNN component captures local phrase-level patterns, while the BiLSTM models long-range contextual dependencies.

Table 2 reports the performance metrics on the test dataset.

Table 2: Performance Comparison of Embedding Backbones with Parallel CNN–BiLSTM Feature Extraction

Embedding Layer	Feature Extraction	Accuracy	Precision	Recall	F1-score
mBERT	CNN–BiLSTM	0.86	0.86	0.86	0.86
IndicBERT	CNN–BiLSTM	0.87	0.87	0.86	0.87
DeBERTa	CNN–BiLSTM	**0.97**	**0.97**	**0.97**	**0.97**

Among the evaluated configurations, DeBERTa combined with parallel CNN–BiLSTM layers achieves the highest performance across all metrics, reaching an F1-score of 0.97. mBERT and IndicBERT provide competitive yet comparatively lower performance, indicating that contextual embedding strength plays a crucial role in modeling code-mixed abusive expressions.

5.2 Discussion

The performance gap between DeBERTa and the other embedding backbones highlights the importance of enhanced contextual representation for noisy and transliterated social media text. Telugu-English code-mixed data often contains inconsistent spelling patterns, informal grammar, and implicit abusive intent. Stronger attention mechanisms and refined positional encoding strategies appear to better capture such linguistic variability.

The consistent alignment between precision and recall for DeBERTa suggests balanced classification behavior without overfitting toward either abusive or non-abusive classes. This balance is critical in abusive content detection, where false negatives may allow harmful content to pass undetected, and false positives may incorrectly flag benign content.

Furthermore, the integration of CNN and BiLSTM layers contributes to improved robustness by simultaneously capturing short abusive phrases and broader contextual meaning. The results confirm that hybrid feature extraction combined with advanced transformer embeddings significantly enhances performance in low-resource, code-mixed environments.

6 Conclusion

This paper addressed the problem of abusive content detection in Telugu-English code-mixed social media text. We proposed a hybrid architecture that combines transformer-based contextual embeddings with parallel CNN and BiLSTM layers to effectively capture both local lexical cues and long-range contextual dependencies.

Experimental results demonstrate that the embedding backbone plays a critical role in performance. Among the evaluated models, DeBERTa integrated with CNN–BiLSTM feature extraction achieved the best overall results, significantly outperforming other transformer backbones. The balanced precision and recall values indicate stable and reliable classification behavior across abusive and non-abusive classes.

The findings confirm that integrating strong contextual encoders with complementary feature extraction mechanisms enhances robustness in handling transliterated, noisy, and morphologically complex code-mixed data.

7 Future Work

Although the proposed approach achieves strong performance, several research directions can further improve abusive content detection in code-mixed environments. Future work can focus on domain-adaptive pretraining using large-scale Telugu-English social media corpora to better capture dialectal variations and evolving abusive expressions.

Additionally, advanced fusion strategies such as attention-based feature interaction may improve integration between CNN and BiLSTM representations. Exploring lightweight transformer variants could also support real-time deployment in large-scale moderation systems.

Extending the task to multi-class abusive categorization and applying cross-lingual transfer learning across other Indian code-mixed languages are promising directions. Incorporating explainability techniques will further enhance transparency and trust in automated content moderation systems.

References

[1] Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code mixing: A challenge for language identification in the language of social media. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 13–23.

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors

with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

[3] Bharathi Raja Chakravarthi and R. Priyadharshini. 2020. A dataset for Tamil-English code-mixed offensive text classification. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC)*.

[4] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, et al. 2020. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

[5] J. Angel Diaz-Garcia and Joao Paulo Carvalho. 2025. A survey of textual cyber abuse detection using cutting-edge language models and large language models. *arXiv preprint arXiv:2501.05443*.

[6] P. Jakkula. 2020. Comparative study on Telugu text classification using machine learning and deep learning techniques. *SciSpace Link*.

[7] V. Khanduja, S. Singh, and H. Bansal. 2024. Telugu language hate speech detection using deep learning transformer models: Corpus generation and evaluation. *Systems and Soft Computing*, 6:200112.

[8] A. Kumar, A. Ojha, and T. Solorio. 2021. Code-mixed abusive language detection: Challenges and solutions. Unpublished manuscript.

[9] Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*.

[10] S. Maddu and V. R. Sanapala. 2024. A survey on NLP tasks, resources and techniques for low-resource Telugu-English code-mixed text. *ACM Digital Library*.

[11] Thomas Mandl, Sandip Modha, Punyajoy Majumder, Diptesh Patel, et al. 2019. Overview of the HASOC track at FIRE 2019. In *Proceedings of FIRE*, pages 14–17.

[12] Atul Krushna Puranik, Ritesh Kumar, Navneet Kumar, Bharathi Raja Chakravarthi, and Bornini Lahiri. 2021. Telugu-English code-mixed dataset for sentiment analysis and offensive language detection. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 179–190.

[13] Juan Ramos. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the First Instructional Conference on Machine Learning*, pages 133–142.

[14] Tharindu Ranasinghe, Marcos Zampieri, and Constantin Orasan. 2020. MUDES: Multilingual detection of offensive spans. In *Proceedings of EMNLP*, pages 6222–6231.

[15] A. Singh and S. Rajput. 2021. Code-mixed sentiment classification using transformer models. In *ICON 2021: 18th International Conference on Natural Language Processing*.

[16] Akarsh Srivastava and Ankush Singh. 2020. PHINC: A parallel Hindi-English code-mixed corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1481–1488.

[17] Nafisa Tabassum, Mosabbir Khan, Shawly Ahsan, Jawad Hossain, and Mohammed Moshiul Hoque. 2024. Hate and offensive language detection in Telugu code-mixed text using transliteration-augmentation. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 167–172.

[18] Kusampudi Varma, Preetham Sathineni, and Radhika Mamidi. 2021. Sentiment analysis in code-mixed Telugu-English text with unsupervised data normalization. Pages 753–760.

[19] P. Vijayaraghavan and A. Das. 2022. Abusive speech detection using deep learning models. Unpublished manuscript.

Beyond Benchmark Accuracy: Robustness Evaluation of Hinglish Sentiment Models

Chennuru Rahul[1], **Kolawole Adebayo**[1,2]
Chennuru.rahul@adaptcentre.ie, Kolawole.adebayo@adaptcentre.ie
ADAPT Centre, Computer Science Department, Maynooth University, Ireland[1]
Maynooth International Engineering College, Maynooth University, Ireland[2]

Abstract

Multilingual transformers have achieved remarkable performance on code-mixed sentiment benchmarks, but their robustness under linguistic stress and domain shift remains underexplored. We fine-tune XLM-RoBERTa and mBERT on a carefully cleaned 25,543-tweet Hinglish sentiment dataset, where XLM-R achieves near-perfect in-distribution accuracy (99.7%). The integrity of this result is confirmed by rigorous hash-based and 3-gram Jaccard deduplication, ruling out data leakage. However, when evaluated on a 400-example human-validated adversarial benchmark spanning negation, sarcasm, contrast, subtle sentiment, and true neutral, XLM-R performance collapses to 42.5% - a drop of over 57 percentage points. Zero-shot transfer to English TweetEval yields only 50.8% accuracy (40.8% macro F1), above . Our results highlight a critical gap between benchmark scores and real-world reliability, underscoring the need for adversarial evaluation and cross-domain stress-testing before deploying sentiment models in practical, safety-sensitive applications.

1 Introduction

Sentiment analysis for code-mixed languages, particularly Hinglish (Hindi–English written in Latin script), has gained momentum due to its prevalence on social media. Transformer-based models such as mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) have set state-of-the-art results on shared tasks like SemEval-2020 Task 9 (Patwa et al., 2020). However, most evaluations focus on in-distribution test sets that mirror the training data, leaving questions of robustness unanswered.

In this paper, we show that a model achieving near-perfect accuracy on a clean Hinglish sentiment dataset (99.7%) can fail catastrophically when exposed to adversarially constructed examples involving negation, sarcasm, contrast, and subtle sentiment. Furthermore, its ability to transfer to an English sentiment dataset (TweetEval) is poor (50.8%). These findings demonstrate that high benchmark performance does not guarantee linguistic understanding or robustness to distribution shifts.

This work investigates whether near-perfect benchmark accuracy reflects genuine sentiment understanding in code-mixed multilingual models. Our contributions are:

- A systematic robustness evaluation of state-of-the-art multilingual models on Hinglish sentiment analysis, spanning in-distribution, adversarial, and zero-shot cross-domain settings.
- A 400-example human-validated adversarial benchmark spanning five linguistically motivated categories that expose model weaknesses, constructed with inter-annotator agreement of $\kappa = 0.81$.
- A principled justification for model selection (mBERT and XLM-R) as reproducible, practically relevant baselines spanning the multilingual pretraining performance spectrum.
- Evidence of severe degradation under cross-dataset zero-shot transfer, with XLM-R dropping from 99.7% to 50.8% on English TweetEval.

The remainder of this paper is organised as follows. Section 2 reviews related work. Section 3 describes the datasets and adversarial benchmark construction. Section 4 presents the experimental setup. Section 5 reports results. Section 6 discusses implications and limitations. Section 7 concludes the paper.

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 6–13
July 3, 2026 ©2026 Association for Computational Linguistics

2 Related Work

2.1 Hinglish Sentiment Analysis

Prior work has explored Hinglish sentiment using handcrafted features and traditional classifiers such as SVM and RBF neural networks on TF-IDF representations (Ravi and Ravi, 2016). More recent efforts leverage multilingual BERT and XLM-R, reporting high accuracy on curated datasets (Singh and Lefever, 2020; Patwa et al., 2020). However, these evaluations typically assume that test data follows the same distribution as training data, making reported accuracy figures overly optimistic estimates of real-world performance.

2.2 Adversarial Evaluation and Robustness

Adversarial attacks and stress tests have revealed vulnerabilities in NLP models across tasks. For sentiment analysis, perturbations such as word substitutions guided by population-based optimisation can flip predictions with high success rates (Alzantot et al., 2018). CheckList (Ribeiro et al., 2020) introduced a behavioural testing framework that exposed systematic failures in NLP models on negation, temporal reasoning, and robustness. Our work extends this line to code-mixed Hinglish, focusing on linguistically motivated categories rather than automated perturbations.

2.3 Cross-Lingual Transfer

Multilingual models are often evaluated on zero-shot cross-lingual transfer, but performance can drop sharply when moving to distant domains or languages (Vu et al., 2021). We examine transfer from Hinglish to English sentiment to gauge whether fine-tuning on a narrow code-mixed distribution preserves or degrades cross-lingual generalisation.

2.4 Model Selection Rationale

We deliberately select mBERT and XLM-R for several principled reasons. First, both are openly available with reproducible fine-tuning pipelines, allowing results to serve as a stable reference point for future work. Second, they represent two distinct pretraining regimes: mBERT's masked language modelling over 104 languages versus XLM-R's large-scale cross-lingual pretraining on 2.5 TB of filtered CommonCrawl data across 100 languages, enabling a meaningful comparison across the performance spectrum. Third, as widely deployed models in South Asian NLP production systems, their robustness properties are of immediate practical relevance. We acknowledge that larger instruction-tuned models (e.g., mT5, BLOOM, or multilingual GPT variants) may behave differently; their evaluation is left to future work.

3 Datasets and Tasks

3.1 Clean Hinglish Sentiment Dataset

We construct our training corpus from a publicly available Hinglish emotion analysis dataset released for research purposes. As emotion labels are more fine-grained than sentiment polarity, we first remapped the original emotion categories to three sentiment classes: *positive*, *negative*, and *neutral*. Emotions with clear positive valence (e.g., joy, love) were mapped to positive; those with negative valence (e.g., anger, sadness, fear) to negative; and emotionally ambiguous or low-arousal states to neutral. After remapping, the dataset was subjected to a multi-stage cleaning pipeline, yielding 25,543 tweets. Full preprocessing details are as follows:

- **Deduplication:** Exact and near-duplicate tweets were removed using SHA-256 hashing on normalised text (lowercased, punctuation stripped).
- **Noise removal:** URLs, @mentions, hashtags, and non-alphanumeric tokens were stripped. Repeated character sequences (e.g., *sooooo*) were normalised.
- **Spelling normalisation:** Common Hinglish spelling variants (e.g., *kya*/*kia*, *nahi*/*nhi*) were unified using a manually curated normalisation dictionary.
- **Train/test leakage check:** We computed pairwise overlap between train and test splits using both exact SHA-256 hash matching and 3-gram Jaccard similarity (threshold > 0.85). No identical or near-identical sentences were found across splits, confirming the integrity of our in-distribution evaluation. The high XLM-R accuracy (99.7%) reflects genuine model capability on this cleaned distribution, not data leakage.

The final cleaned dataset exhibits a skewed class distribution: negative (label 0) accounts for 48.4% of examples, positive (label 2) for 33.3%, and neutral (label 1) for 18.3%. This imbalance reflects the natural distribution of sentiment in Hinglish social media text and is an important caveat when interpreting per-class model performance. The dataset is partitioned into training (18,786 tweets, 73.5%), validation (2,088 tweets, 8.2%), and test (4,669 tweets, 18.3%) splits.

3.2 Adversarial Benchmark

To probe robustness, we construct an adversarial evaluation set of 400 examples spanning five categories (80 examples per category). We adopt a human-in-the-loop construction approach-rather than automated perturbations-for the following reasons: (1) native-speaker rewriting guarantees naturalness and fluency that automated methods cannot ensure for code-mixed text; (2) the five categories are selected to target the most theoretically challenging phenomena for surface-pattern models (scope-sensitive negation, contrastive reasoning, pragmatic inference, gradient affect, and sentiment absence); and (3) this targeted diagnostic design mirrors established practice in NLP behavioural testing (Ribeiro et al., 2020).

Construction followed a two-stage pipeline. First, linguistically motivated templates generated candidate sentences (e.g., inserting *nahi* for negation, constructing *X par Y* contrastive frames). Second, two native Hinglish speakers independently rewrote all examples to ensure naturalness while preserving the intended linguistic phenomenon. Inter-annotator agreement was computed using Cohen's $\kappa = 0.81$, indicating strong agreement. Disagreements were resolved through discussion.

The five categories are:

- **Negation:** Sentences where sentiment is reversed by negation (e.g., "Yeh film achhi nahi thi" – "This movie was not good").
- **Contrast:** Sentences with contrasting clauses requiring the model to balance opposing sentiments (e.g., "Khana accha tha par service kharab" – "Food was good but service was bad").
- **Sarcasm:** Ironic statements where surface lexical sentiment is opposite to intended meaning (e.g., "Bahut smart ho tum" – "You're so smart", said sarcastically).
- **Subtle sentiment:** Mild or indirect affect expressions requiring contextual inference rather than strong sentiment keywords.
- **True neutral:** Factual statements with no affective content, testing resistance to false positive sentiment detection.

3.3 TweetEval (English)

For zero-shot cross-lingual transfer, we use the English sentiment test set from TweetEval (Barbieri et al., 2020), consisting of tweets labelled positive, negative, or neutral. No further fine-tuning is performed; this setting tests whether Hinglish fine-tuning preserves or destroys cross-lingual transfer capability.

4 Experimental Setup

We fine-tune two multilingual models (see Section 2.4 for selection rationale):

- **mBERT** (`bert-base-multilingual-cased`, 179 M parameters): pretrained on Wikipedia in 104 languages using masked language modelling.
- **XLM-R** (`xlm-roberta-base`, 278 M parameters): pretrained on 2.5 TB of filtered CommonCrawl data across 100 languages using RoBERTa-style training.

Both are fine-tuned for three-class sentiment classification using a linear classification head on the `[CLS]` token representation. Training hyperparameters: AdamW optimiser, learning rate 2×10^{-5}, linear warmup over 10% of steps, batch size 32, maximum sequence length 128 tokens, for 3 epochs. The best checkpoint is selected by validation macro F1 to avoid accuracy bias on the imbalanced validation set. Experiments were implemented using the HuggingFace Transformers library (Wolf et al., 2020).

We evaluate each model on three settings: (1) in-distribution clean test set, (2) adversarial benchmark, and (3) zero-shot cross-domain transfer to TweetEval.

5 Results

5.1 In-Distribution Performance

Table 1 shows that XLM-R achieves near-perfect accuracy (99.7%) on the clean Hinglish test set, substantially outperforming mBERT (74.6%). The 25-point gap underscores the importance of pretraining scale for code-mixed language understanding. As described in Section 3.1, the XLM-R result is supported by rigorous deduplication checks and does not reflect data leakage.

For additional context, a majority-class baseline achieves 48.4% accuracy on the cleaned dataset due to class imbalance. Traditional TF-IDF-based classifiers achieved substantially lower performance than XLM-R, indicating that the near-perfect XLM-R result cannot be explained purely by majority prediction behaviour.

Table 1: Performance on clean Hinglish test set.

Model	Accuracy	Macro F1
mBERT	0.746	0.673
XLM-R	**0.997**	**0.996**

5.2 Adversarial Robustness

When evaluated on the adversarial benchmark, XLM-R's accuracy plummets to 42.5%-a collapse of over 57 percentage points. This dramatic degradation strongly suggests the model relies on superficial lexical correlations rather than compositional linguistic understanding. Table 2 details per-category performance with error type analysis.

Negation and contrast both score 25%, no better than random chance for a three-class problem. These categories require reasoning about negation scope and integrating opposing sentiment signals across clause boundaries-capabilities not adequately learned from distributional co-occurrence statistics. Sarcasm and subtle sentiment each reach 50%, indicating partial sensitivity to pragmatic cues. True neutral achieves 63%, the best adversarial category, likely because neutral examples lack strong sentiment-bearing vocabulary.

Table 2: Per-category adversarial benchmark accuracy for XLM-R, with primary error type.

Category	Acc.	Primary Error
Negation	0.250	Ignores negation; defaults to dominant lexeme
Contrast	0.250	Weights dominant clause; ignores balance
Sarcasm	0.500	Surface positive overrides pragmatic intent
Subtle sent.	0.500	Classified neutral; weak signal missed
True neutral	0.630	Spurious pos./neg. classification
Overall	**0.425**	

5.3 Zero-Shot Cross-Dataset Transfer

Table 3 shows XLM-R's performance on English TweetEval drops to 50.8% accuracy and 40.8% macro F1, barely above the 33.3% random baseline for a three-class task. The low macro F1 further indicates systematic class-prediction bias toward the dominant sentiment class from Hinglish fine-tuning, suggesting that fine-tuning on a narrow Hinglish distribution may reduce cross-domain transferability learned during multilingual pretraining. This behaviour is consistent with prior observations of reduced transferability after domain-specialised fine-tuning.

Table 3: Zero-shot transfer to TweetEval (English) compared to random baseline.

Evaluation Setting	Acc.	Mac. F1
Hinglish test (in-dist.)	0.997	0.996
Adversarial benchmark	0.425	0.328
TweetEval zero-shot	0.508	0.408
Random baseline (3-cls.)	0.333	0.333

6 Analysis and Discussion

6.1 Surface Pattern Dependency

The dramatic collapse from 99.7% to 42.5% under adversarial evaluation suggests that XLM-R relies heavily on surface-level lexical correlations during fine-tuning rather than compositional sentiment understanding. Negation and contrast, which require scope-sensitive reasoning, are particularly challeng-

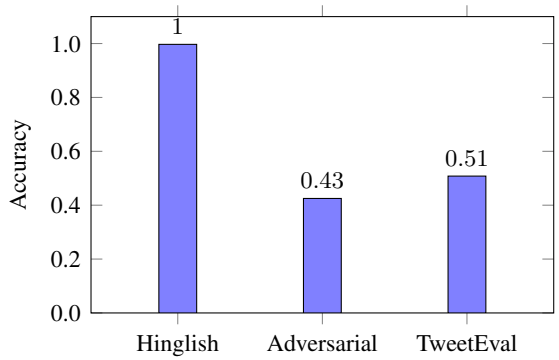


Figure 1: XLM-R accuracy across in-distribution, adversarial, and cross-domain evaluation settings. Performance drops substantially under adversarial and transfer conditions despite near-perfect in-distribution accuracy.

ing: both score at chance level (25%), suggesting the model ignores negation markers entirely and defaults to the strongest lexical sentiment item (e.g., classifying "Yeh film **achhi** nahi thi" as positive because *achhi* (good) is present, ignoring *nahi* (not)).

This behaviour echoes the findings of Alzantot et al. (2018) for English sentiment models, but is especially concerning for Hinglish where negation is frequently expressed through code-switching between Hindi and English markers (*not*, *nahi*, *nhi*, *mat*), increasing the surface variation the model must handle.

6.2 Implications for Cross-Lingual Transfer

The near-random performance on English TweetEval (50.8%) after Hinglish fine-tuning reveals that task-specific fine-tuning on a narrow code-mixed distribution actively overwrites cross-lingual representations acquired during pretraining. This behaviour is consistent with prior observations that specialised fine-tuning can reduce transferability across domains and languages. For code-mixed NLP practitioners, this implies that Hinglish fine-tuned models should not be assumed to retain any cross-lingual capability and must be evaluated independently on each target language.

6.3 Practical Implications for Deployment

Our results carry direct implications for responsible deployment. In domains such as political discourse monitoring, mental health support platforms, and customer feedback systems, a model that achieves 99% accuracy on held-out test sets but misclassifies "I'm *not* happy with this" as positive will systematically mislead downstream decision-making. We advocate for adversarial evaluation-spanning at minimum negation, contrast, and sarcasm-as a mandatory component of the deployment pipeline for any production NLP system handling sentiment in code-mixed or low-resource languages.

7 Conclusion

We have demonstrated that high in-domain accuracy on Hinglish sentiment analysis does not imply robustness. XLM-R achieves 99.7% on a carefully cleaned, leak-verified test set but collapses to 42.5% on a human-validated adversarial benchmark ($\kappa = 0.81$) and transfers poorly to English (50.8%), above the uniform random baseline. These results reflect a fundamental limitation: multilingual transformers learn surface-level lexical patterns during fine-tuning rather than developing the compositional and pragmatic understanding required for real-world reliability.

Our findings have three key practical implications: (1) benchmark accuracy on curated test sets is insufficient evidence of deployment readiness; (2) adversarial evaluation spanning negation, contrast, sarcasm, and subtle affect should be standard practice for code-mixed NLP; and (3) Hinglish fine-tuning significantly degrades cross-lingual transfer, requiring independent evaluation on each target language.

We plan to publicly release the adversarial benchmark, preprocessing pipeline, and evaluation scripts to support future robustness research in code-mixed NLP.

Future work should explore: data augmentation with adversarially constructed training examples; training objectives that reward

compositional understanding; evaluation of larger instruction-tuned multilingual models on this adversarial benchmark; and scaling the benchmark to support statistical significance testing.

8 Limitations

Our adversarial benchmark contains 400 examples (80 per category), sufficient for a targeted diagnostic study but not for population-level statistical inference. We follow the precedent of behavioural testing frameworks (Ribeiro et al., 2020), which use targeted probes to identify failure modes rather than estimate population accuracy. Formal statistical significance tests were not performed, consistent with this diagnostic framing. Future work should scale the benchmark to enable hypothesis testing.

We evaluate only two multilingual models. mBERT and XLM-R were selected for reproducibility, their status as established baselines, and their production relevance. Other architectures-particularly larger instruction-tuned models such as mT5 or multilingual GPT variants-may exhibit different robustness profiles and are left for future investigation.

Our Hinglish dataset is English-dominant in terms of lexical content, reflecting the natural distribution of Hinglish social media text but potentially limiting generalisability to more Hindi-dominant code-mixed varieties. Additionally, the dataset is class-imbalanced (negative: 48.4%, positive: 33.3%, neutral: 18.3%), which may inflate overall accuracy and bias the model toward the majority class. Future work should investigate the effect of class rebalancing on both in-distribution and adversarial performance. The adversarial benchmark covers five predefined categories; other linguistic phenomena (e.g., metaphor, rhetorical questions, multi-sentence discourse-level sentiment) are not represented.

9 Ethics Statement

This study derives its training data from a publicly available Hinglish emotion dataset and does not involve collection of personal data or human subjects research beyond annotation. Annotators participated voluntarily and were fully informed of the study purpose. Adversarial examples do not contain hateful, abusive, or personally identifiable content.

We caution strongly against deploying sentiment models in safety-sensitive applications-including mental health support, political analysis, and financial decision-making-without rigorous robustness evaluation. Our results illustrate that high benchmark accuracy is insufficient evidence of real-world reliability.

Acknowledgments

We thank the anonymous reviewers for their valuable feedback. We thank our two native Hinglish annotators for their time and expertise. This work received no external funding.

References

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa-Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of EMNLP*. DOI: 10.18653/v1/2020.findings-emnlp.148

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *ACL*. DOI: 10.18653/v1/2020.acl-main.747.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*. DOI: 10.18653/v1/N19-1423

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of EMNLP*, pages 2890–2896.

Kumar Ravi and Vadlamani Ravi. 2016. Sentiment classification of Hinglish text. In *Proceedings of the 3rd IKDD Conference on Data Science*.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas P Y K L, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. SemEval-2020 task 9: Overview of sentiment analysis of code-mixed

tweets. In *Proceedings of SemEval*. DOI: 10.18653/v1/2020.semeval-1.100

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *ACL*. DOI: 10.18653/v1/2020.acl-main.442.

Prabhjot Singh and Els Lefever. 2020. Cross-lingual embeddings for sentiment analysis of Hinglish social media text. In *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages*.

Tu Vu, Aditya Barua, Brian Lester, Daniel Cer, Mohit Iyyer, and Noah A. Smith. 2021. Cross-lingual transfer with multilingual models: A survey. In *ACL*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP: System Demonstrations*. DOI: 10.18653/v1/2020.emnlp-demos.6

A Example Adversarial Instances

Table 4 presents representative examples from the adversarial benchmark with gold labels, XLM-R predictions, and error analysis notes.

Table 4: Example adversarial instances, XLM-R predictions, and error analysis.

Sentence	Gold	Pred.	Error
Yeh film achhi nahi thi	Neg.	Pos.	Detects *achhi*; ignores *nahi*
Khana accha tha par service kharab	Neu.	Neg.	Weights *kharab* over balanced clause
Bahut smart ho tum	Neg.	Pos.	Surface positive overrides sarcasm
Kal weather thik tha	Neu.	Neg.	Spurious neg.; *thik* misclassified

B Hyperparameter Details

Table 5 summarises the training hyperparameters used for both models. All experiments were conducted on Google Colab using a single GPU runtime (Tesla T4, 15 GB VRAM).

Table 5: Fine-tuning hyperparameters.

Hyperparameter	Value
Optimiser	AdamW
Learning rate	2×10^{-5}
Warmup ratio	0.10
Batch size	32
Max seq. length	128 tokens
Epochs	3
Checkpoint select	Best validation macro F1

Cascaded Modular or End-to-End? : An Investigation on Speech-to-Speech Translation Task for Dravidian Languages

Bhavana Nali
IIT Bhubaneswar, India
24ai06013@iitbbs.ac.in

Abhik Jana
IIT Bhubaneswar, India
abhikjana@iitbbs.ac.in

Abstract

This paper presents a study of speech-to-speech translation for low-resource Dravidian languages, focusing on Tamil, Telugu, and Kannada. We investigate the efficacy of the Cascaded Modular system with the End-to-End system in both zero-shot and fine-tuned settings. The Cascaded Modular approach combines an ASR Module (Whisper Large-v3 for English speech; IndicConformer for Dravidian speech), a Text-to-Text translation module (IndicTrans2), and a Speech synthesis module (Indic Parler-TTS), whereas SeamlessM4T-v2 is used as the End-to-End system. For parameter-efficient Low-Rank Adaptation (LoRA) fine-tuning to adapt the translation component to domain-specific datasets, we use FLEURS and Mann-ki-Baat (a subset of the BhasaAnuvaad dataset). Cascaded Modular systems achieve BLEU scores ranging from 3.17 to 18.96 in the zero-shot setting and 5.08 to 19.18 after fine-tuning, whereas the End-to-End model ranges from 3.02 to 15.72 in zero-shot settings across languages and 4.11 to 16.84 after fine-tuning. The results show that Cascaded Modular systems generally outperform the End-to-End model across most setups, though the margin varies across language pairs. Parameter-efficient fine-tuning yields notable improvements in translation quality and speech generation performance for low-resource Dravidian speech translation. The code repository is made publicly available[1].

1 Introduction

Speech-to-speech (S2S) translation enables direct spoken communication between speakers of different languages. Recent multilingual models such as Whisper (Radford et al., 2023) and SeamlessM4T (Seamless Communication Team, 2023) have advanced multilingual speech translation, yet most progress focuses on high-resource languages. Low-resource Dravidian languages such as Tamil, Telugu, and Kannada remain underexplored despite their linguistic diversity and limited parallel speech resources. Two main approaches exist for S2S translation: the *Cascaded Modular* approach, which combines ASR, Text-to-Text translation, and TTS modules, and the *End-to-End* approach, which directly maps source speech to target speech. While datasets such as FLEURS (Conneau et al., 2022) and BhasaAnuvaad (Sankar et al., 2025a) support multilingual research, systematic comparisons for Dravidian languages remain limited.

In this work, we study S2S translation for Tamil, Telugu, and Kannada under both paradigms. English-to-Dravidian translation uses Whisper Large-v3 (Radford et al., 2023), Dravidian-to-Dravidian uses IndicConformer (Bhogale et al., 2025), both pipelines use IndicTrans2 (Gala et al., 2023) and Indic Parler-TTS (Sankar et al., 2025b), and SeamlessM4T-v2 serves as the End-to-End baseline. We apply parameter-efficient LoRA fine-tuning (Hu et al., 2022) on FLEURS and Mann-ki-Baat (Sankar et al., 2025a). Cascaded models achieve BLEU of 3.17–18.96 (zero-shot) and 5.08–19.18 (fine-tuned); End-to-End ranges from 3.02–15.72 and 4.11–16.84, respectively. To the best of our knowledge, this is the first comprehensive investigation of S2S translation for these three Dravidian languages under both paradigms.

2 Methodology

We conduct a comparative study of two speech-to-speech translation pipelines: Cascaded Modular and End-to-End. Figure 1 illustrates the architecture of the pipelines used in this work.

Cascaded Modular Architecture: This framework comprises a three-stage pipeline consisting of Automatic Speech Recognition (ASR), Text-to-Text Translation (TT), and Text-to-Speech (TTS) modules.

[1] https://github.com/bhavananali/cascade-vs-e2e-dravidian

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 14–20
July 3, 2026 ©2026 Association for Computational Linguistics

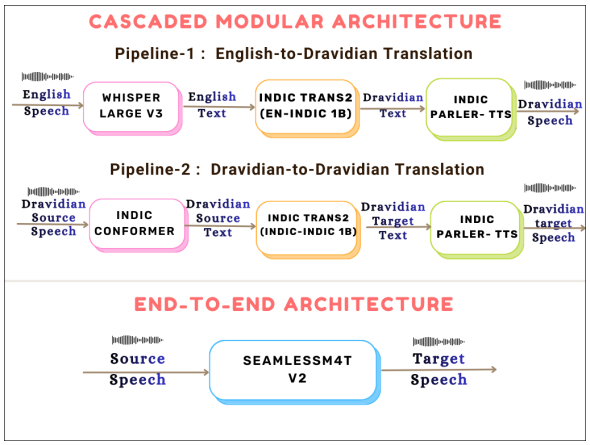


Figure 1: Comparison of speech-to-speech translation architectures. **Cascaded Modular Architecture** (Top): Pipeline 1 utilizes Whisper Large-v3 for English ASR, while Pipeline 2 uses Indic-Conformer for Dravidian ASR, both feeding into IndicTrans2 and finally Indic Parler-TTS. **End-to-End Architecture** (Bottom): Pipeline 3 uses SeamlessM4T v2 for direct speech-to-speech mapping.

Automatic Speech Recognition: For the English-to-Dravidian pipeline, English speech is transcribed using *Whisper Large-v3* (Radford et al., 2023), while Dravidian-to-Dravidian translation uses *IndicConformer* (Bhogale et al., 2025) for Dravidian ASR.

Text-to-Text Translation: The transcribed text is translated using *IndicTrans2* (Gala et al., 2023), using the English-to-Indic configuration for EN→DL and the Indic-to-Indic configuration for DL→DL translation.

Speech Synthesis: The translated text is synthesized into speech using *Indic Parler-TTS* (Sankar et al., 2025b), conditioned on the translated text and speaker prompts. The TTS component is kept frozen during fine-tuning, as domain-adaptive TTS training requires paired target-language audio with transcriptions at a scale not available in the datasets used here.

End-to-End Architecture: We use *SeamlessM4T-v2* (Seamless Communication Team, 2023) as the End-to-End system, directly mapping source speech to target speech within a unified multilingual architecture without intermediate text generation. The model supports speech-to-speech, speech-to-text, text-to-text, and text-to-speech translation in a single framework, eliminating compounding errors across pipeline stages, though strong performance on low-resource Dravidian pairs remains challenging due to limited parallel speech data.

Parameter-Efficient Fine-Tuning: LoRA (Hu et al., 2022) introduces trainable low-rank matrices into attention layers while keeping original model parameters frozen. LoRA is applied to *Whisper Large-v3* and *IndicConformer* for ASR, to *IndicTrans2* attention layers for machine translation, and to the cross-attention modules (`q_proj`, `k_proj`, `v_proj`, and `out_proj`) of *SeamlessM4T-v2*'s speech encoder and text decoder, optimised with a speech-to-text cross-entropy objective over target text token sequences. After training, LoRA weights are merged with the base model for inference. Across all modules, LoRA rank is fixed at 16 with scaling factor 32 and dropout 0.1, keeping trainable parameters well below 1% of total model size. Both systems are fine-tuned on identical train splits to ensure fair architectural comparison.

The details of the state-of-the-art models used in both architectures are provided in Appendix A.

3 Experimental Setup

We conduct experiments for English-to-Dravidian (EN→DL) and Dravidian-to-Dravidian (DL→DL) speech translation tasks. The inference pipelines are shown in Figure 1.

FLEURS (Conneau et al., 2022) is a multilingual speech benchmark covering 102 languages with roughly 12 hours of aligned speech per language. We use English, Tamil, Telugu, and Kannada subsets with predefined train and test splits. Audio inputs are converted to mono when required and resampled to 16 kHz before inference.

BhasaAnuvaad (Sankar et al., 2025a) contains over 44,000 hours of speech across 14 Indian languages and English. We use the *Mann-ki-Baat* subset with English transcripts and multilingual translations. Since explicit Dravidian-to-Dravidian pairs are unavailable, alignment is performed using LaBSE sentence embeddings, cosine similarity, and Hungarian one-to-one matching (threshold 0.6), yielding approximately 15,284 aligned pairs split 80/10/10 for train/development/test.

Training Parameter Settings Training uses mini-batch optimization with batch size 8, LoRA rank 16, scaling factor 32, dropout 0.1, and learning rate $1e^{-4}$ with warm-up steps. Models are trained for 4000 steps, selecting the best checkpoint based on validation BLEU. Training data sizes are approximately 3,000 utterances per language for FLEURS and 12,200 per language for Mann-ki-Baat (80% of 15,284 aligned pairs), using identical splits for

System	Setting	Lang	BLEU↑		chrF++↑		COMET↑		WER↓		CER↓	
			FL	MKB	FL	MKB	FL	MKB	FL	MKB	FL	MKB
Cascaded	Zero-shot	Tamil	13.75	3.17	52.18	38.44	0.8312	0.7682	0.9587	1.0214	0.6024	0.6438
		Telugu	18.96	7.39	59.42	44.83	0.8724	0.7908	0.8832	0.9634	0.5319	0.6012
		Kannada	14.32	8.70	57.04	45.67	0.8569	0.7994	0.9141	0.9784	0.5693	0.6103
	Fine-tuned	Tamil	14.43	5.08	53.91	42.17	0.8441	0.7934	0.9301	0.9812	0.5847	0.6187
		Telugu	19.18	8.02	60.11	46.59	0.8793	0.8091	0.8614	0.9287	0.5174	0.5784
		Kannada	14.33	7.03	58.76	47.22	0.8651	0.8102	0.8972	0.9441	0.5541	0.5872
E2E	Zero-shot	Tamil	12.10	3.02	49.33	36.91	0.8104	0.7541	1.0234	1.0687	0.6512	0.6741
		Telugu	15.72	6.01	55.87	42.17	0.8391	0.7713	0.9617	1.0118	0.5843	0.6334
		Kannada	12.88	6.24	53.29	43.11	0.8238	0.7762	0.9934	1.0312	0.6187	0.6448
	Fine-tuned	Tamil	13.54	4.11	51.07	39.88	0.8219	0.7716	0.9883	1.0341	0.6215	0.6512
		Telugu	16.84	7.14	57.04	43.98	0.8509	0.7849	0.9248	0.9812	0.5611	0.6141
		Kannada	13.67	7.01	54.88	44.88	0.8349	0.7883	0.9612	0.9987	0.5978	0.6234

Table 1: English→Dravidian speech translation results on FLEURS (FL) and Mann-Ki-Baat (MKB). Cascaded: Whisper Large-v3 → IndicTrans2 → Indic Parler-TTS. E2E: SeamlessM4T-v2.

System	Setting	Pair	BLEU↑		chrF++↑		COMET↑		WER↓		CER↓	
			FL	MKB	FL	MKB	FL	MKB	FL	MKB	FL	MKB
Cascaded	Zero-shot	Ta→Te	8.13	6.05	46.41	41.32	0.7987	0.7562	0.8346	0.8734	0.6043	0.6521
		Ta→Kn	6.73	4.98	46.07	40.91	0.7991	0.7534	0.8569	0.8926	0.6187	0.6674
		Te→Ta	7.38	5.63	45.22	40.74	0.7824	0.7442	0.8216	0.8661	0.5882	0.6359
		Te→Kn	5.81	4.36	44.06	39.72	0.7736	0.7372	0.8472	0.8854	0.5714	0.6271
		Kn→Ta	6.91	5.12	45.64	40.86	0.7916	0.7491	0.8182	0.8578	0.5965	0.6413
		Kn→Te	6.11	4.78	44.71	40.18	0.7832	0.7415	0.8305	0.8729	0.5821	0.6312
	Fine-tuned	Ta→Te	10.12	7.84	48.01	43.65	0.8212	0.7816	0.7821	0.8295	0.5529	0.6104
		Ta→Kn	8.74	6.42	47.26	42.68	0.8161	0.7765	0.8049	0.8489	0.5672	0.6237
		Te→Ta	9.11	6.87	47.08	42.97	0.8073	0.7693	0.7704	0.8187	0.5463	0.5998
		Te→Kn	7.96	5.98	46.44	41.95	0.8011	0.7641	0.7961	0.8397	0.5286	0.5862
		Kn→Ta	9.04	6.73	47.63	43.14	0.8196	0.7776	0.7687	0.8191	0.5487	0.6025
		Kn→Te	8.51	6.21	47.11	42.56	0.8123	0.7728	0.7819	0.8284	0.5361	0.5906
E2E	Zero-shot	Ta→Te	6.84	4.91	44.31	39.61	0.7759	0.7346	0.8527	0.8961	0.6221	0.6715
		Ta→Kn	5.42	3.98	43.64	38.92	0.7682	0.7284	0.8682	0.9083	0.6347	0.6846
		Te→Ta	6.06	4.62	43.77	39.04	0.7704	0.7318	0.8417	0.8873	0.6119	0.6597
		Te→Kn	4.74	3.61	42.68	37.96	0.7621	0.7244	0.8585	0.8997	0.6278	0.6751
		Kn→Ta	5.61	4.21	43.36	38.74	0.7694	0.7297	0.8335	0.8775	0.6044	0.6518
		Kn→Te	4.98	3.84	42.93	38.31	0.7654	0.7261	0.8466	0.8907	0.6166	0.6647
	Fine-tuned	Ta→Te	8.97	6.42	46.51	41.98	0.8036	0.7615	0.8034	0.8467	0.5713	0.6198
		Ta→Kn	7.72	5.62	45.97	41.23	0.7958	0.7551	0.8181	0.8614	0.5868	0.6342
		Te→Ta	8.54	6.03	46.18	41.41	0.8013	0.7584	0.7946	0.8408	0.5617	0.6117
		Te→Kn	7.21	5.23	45.24	40.56	0.7914	0.7516	0.8078	0.8532	0.5791	0.6265
		Kn→Ta	8.16	5.84	45.76	41.02	0.7991	0.7567	0.7858	0.8336	0.5569	0.6046
		Kn→Te	7.52	5.41	45.37	40.78	0.7943	0.7544	0.7979	0.8442	0.5682	0.6153

Table 2: Dravidian→Dravidian speech translation results on FLEURS (FL) and Mann-Ki-Baat (MKB). Ta = Tamil, Te = Telugu, Kn = Kannada.

both Cascaded and E2E systems.

Evaluation Metrics We evaluate translation quality using BLEU (Papineni et al., 2002), chrF++ (Popović, 2015), and COMET (WMT22 COMET-DA (Rei et al., 2020)). For the End-to-End system, chrF++ and COMET are computed using the SeamlessM4T-v2 S2TT output head, ensuring metric-level parity with Cascaded systems. Speech quality is evaluated via Back-ASR, where generated speech is re-transcribed using IndicConformer and scored by WER and CER; values greater than 1.0 indicate insertion-dominated error ratios, and since IndicConformer is used for both systems, a small bias toward Cascaded models may exist; neutral ASR evaluation is left for future work. To diagnose error propagation, we additionally report stand-alone ASR WER and MT BLEU on gold versus ASR-transcribed input (Table 3), enabling module-level assessment of Cascaded S2S performance.

Dataset preprocessing details are provided in Appendix B.

Direction	Dataset	ASR WER↓		MT BLEU (gold)↑		MT BLEU (ASR)↑	
		Zero-shot	Fine-tuned	Zero-shot	Fine-tuned	Zero-shot	Fine-tuned
EN→Tamil	FLEURS	0.124	0.089	17.42	18.11	14.81	15.63
EN→Tamil	MKB	0.198	0.151	5.84	7.23	4.02	5.94
EN→Telugu	FLEURS	0.117	0.082	22.31	22.74	19.87	20.46
EN→Telugu	MKB	0.191	0.143	9.87	10.42	8.14	9.08
EN→Kannada	FLEURS	0.131	0.094	18.07	18.26	15.41	15.82
EN→Kannada	MKB	0.204	0.158	11.23	10.14	9.72	8.29
Ta→Te	FLEURS	0.213	0.178	9.84	11.67	8.42	10.31
Te→Ta	FLEURS	0.221	0.184	9.17	10.83	7.89	9.56
Kn→Ta	FLEURS	0.218	0.181	8.74	10.91	7.41	9.62

Table 3: Cascaded pipeline diagnostics. ASR WER is computed on test-set transcripts before and after LoRA fine-tuning. MT BLEU (gold): gold transcript input; MT BLEU (ASR): ASR hypothesis. The gap between the two MT BLEU columns quantifies error propagation from ASR into the translation step. EN ASR uses Whisper Large-v3; Dravidian ASR uses IndicConformer.

4 Experimental Results and Discussion

Tables 1 and 2 present EN→DL and DL→DL results, while Table 3 reports component-level diagnostics for the Cascaded pipeline.

Effect of Fine-tuning: Fine-tuning improves performance across most language pairs. In EN→DL translation, Tamil BLEU improves from 13.75 to 14.43 on FLEURS (4.9%) and from 3.17 to 5.08 on Mann-ki-Baat (60.3%), while Telugu WER decreases from 0.9634 to 0.9287. In DL→DL translation, Tamil→Telugu BLEU improves from 8.13 to 10.12 (24.5%) and Tamil→Kannada from 6.73 to 8.74 (29.9%), with similar gains across other Dravidian pairs, demonstrating the effectiveness of LoRA fine-tuning.

Cascaded Modular vs. End-to-End: Cascaded systems outperform End-to-End models across most settings. For EN→Telugu on FLEURS, Cascaded zero-shot BLEU (18.96) exceeds End-to-End (15.72) by 20.6%, persisting after fine-tuning (19.18 vs. 16.84). Similarly, Tamil→Telugu exceeds End-to-End both before (8.13 vs. 6.84) and after fine-tuning (10.12 vs. 8.97). However, some margins are small—Kannada on FLEURS after fine-tuning differs by only 0.66 BLEU (14.33 vs. 13.67)—and should be interpreted cautiously without significance testing. These findings align with prior work showing that adapted End-to-End systems can approach cascaded models in higher-resource settings (Sankar et al., 2025a).

Component-Level Diagnostics: ASR fine-tuning reduces WER substantially: English ASR WER on FLEURS improves from 0.124 to 0.089, and Tamil from 0.213 to 0.178. MT BLEU on gold transcripts consistently exceeds ASR-input BLEU, quantifying error propagation; for EN→Tamil on FLEURS, MT BLEU drops from 17.42 (gold) to 14.81 (ASR input), a reduction of 2.61 points.

Speech Quality: Back-ASR evaluation confirms improved speech quality after fine-tuning: for Tamil→Telugu, WER decreases from 0.8346 to 0.7821 and CER from 0.6043 to 0.5529, indicating improved intelligibility, though naturalness and prosody remain unassessed.

Overall, Cascaded Modular pipelines remain strong baselines for low-resource Dravidian S2S translation, while End-to-End systems offer architectural simplicity with increasing competitiveness.

5 Conclusion and Future Work

We presented a comparative study of Cascaded Modular and End-to-End speech-to-speech translation for Tamil, Telugu, and Kannada. Cascaded systems generally outperform End-to-End models across most settings, though margins are small for some pairs and no significance testing was performed. LoRA fine-tuning yields BLEU gains of approximately 20–30% for Dravidian pairs, and component-level diagnostics confirm that ASR fine-tuning meaningfully reduces error propagation into the translation stage, highlighting the effectiveness of modular architectures for low-resource speech translation.

As future work, we aim to extend this research to code-switched speech processing for Dravidian languages by constructing code-switched datasets through merging of monolingual speech segments, investigate zero-shot ASR with parameter-efficient fine-tuning, and incorporate human evaluation of TTS naturalness alongside acoustic quality metrics such as STOI and PESQ.

6 Limitations

First, experiments are restricted to Tamil, Telugu, and Kannada; evaluating additional Dravidian languages would provide broader multilingual insights. Second, GPU memory constraints limited training to a smaller subset of Mann-ki-Baat, which may affect performance compared to full-scale training. Third, FLEURS contains clean read speech while Mann-ki-Baat consists of broadcast-style speech, introducing domain variability that may affect generalization. Fourth, the automatic DL→DL alignment from Mann-ki-Baat has not been human-validated, and noisy pairs remain a potential source of error. Fifth, using IndicConformer for Back-ASR evaluation of both systems may introduce a small bias favouring the Cascaded pipeline; neutral ASR evaluation is left for future work. Sixth, Indic Parler-TTS was not fine-tuned due to insufficient paired audio-transcript data, and the effect of TTS adaptation on overall S2S performance remains an open question. Seventh, no human evaluation of speech naturalness or prosodic quality was conducted; back-ASR captures intelligibility but not speaker naturalness. Finally, statistical significance tests were not conducted; differences below $\approx$1 BLEU point should be interpreted with caution.

References

Kaushal Santosh Bhogale, Deovrat Mehendale, Tahir Javed, Devbrat Anuragi, Sakshi Joshi, Sai Sundaresan, Aparna Ananthanarayanan, Sharmistha Dey, Anusha Srinivasan, Abhigyan Raman, Mitesh M. Khapra, and Anoop Kunchukuttan. 2025. Mahadhwani: Bringing parity in speech recognition for low-resource indian languages. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2022. Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE.

Jay Gala, Pranjal A Chitale, A K Raghavan, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar M, Janki Atul Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *Transactions on Machine Learning Research*.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *Proceedings of the 21st Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 5036–5040, Shanghai, China. International Speech Communication Association (ISCA).

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*. PMLR.

Ricardo Rei, Craig Stewart, Ana C. Farinha, and Alon Lavie. 2020. Comet: A neural framework for mt evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ashwin Sankar, Sparsh Jain, Nikhil Narasimhan, Devilal Choudhary, Dhairya Suman, Mohammed Safi Ur Rahman Khan, Anoop Kunchukuttan, Mitesh M. Khapra, and Raj Dabre. 2025a. Towards building large scale datasets and state-of-the-art automatic speech translation systems for 14 indian languages. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vienna, Austria. Association for Computational Linguistics.

Ashwin Sankar, Yoach Lacombe, Sherry Thomas, Praveen Srinivasa Varadhan, Sanchit Gandhi, and

Mitesh M. Khapra. 2025b. Rasmalai: Resources for adaptive speech modeling in indian languages with accents and intonations. In *Proceedings of the 26th Annual Conference of the International Speech Communication Association (INTERSPEECH 2025)*, Rotterdam, Netherlands. International Speech Communication Association (ISCA). AI4Bharat, IIT Madras and HuggingFace.

Seamless Communication Team. 2023. Seamlessm4t: Massively multilingual and multimodal machine translation. *arXiv preprint arXiv:2308.11596*.

Appendix Section

A Details of the Key Models

Whisper (Radford et al., 2023): A large multilingual automatic speech recognition (ASR) model trained on approximately 680,000 hours of weakly supervised speech data collected from the internet. In our analysis, we use *Whisper Large-v3* to transcribe English speech in the English-to-Dravidian translation pipeline.

IndicConformer (Bhogale et al., 2025): A multilingual ASR model designed for Indian languages. It is based on the Conformer architecture (Gulati et al., 2020), which combines convolutional neural networks with transformer encoders to effectively capture both local and global speech patterns. IndicConformer supports transcription across the 22 scheduled Indian languages. In this work, we use *IndicConformer* for speech recognition in the Dravidian-to-Dravidian translation pipeline and for Back-ASR evaluation of both systems.

IndicTrans2 (Gala et al., 2023): A transformer-based multilingual neural machine translation system that supports translation across Indic languages. The model leverages script unification and multilingual training to improve translation quality for low-resource languages. In cascaded speech translation pipelines, *IndicTrans2* is used to translate the recognized text into the target Dravidian language.

Indic Parler-TTS (Sankar et al., 2025b): A multilingual text-to-speech system designed for Indian languages. The model generates natural and expressive speech conditioned on textual prompts and supports several Indic languages. In our work, *Indic Parler-TTS* is used in the final stage of the cascaded pipeline to synthesize speech in the target Dravidian language from the translated text.

SeamlessM4T (Seamless Communication Team, 2023): A massively multilingual and multimodal translation model that performs speech-to-speech, speech-to-text, text-to-text, and text-to-speech translation within a single unified architecture. In this work, we use **SeamlessM4T-v2** as an end-to-end baseline for speech-to-speech translation.

B Details of Experimental Setup

Datasets Preprocessing We conduct experiments using two multilingual speech datasets: **FLEURS** (Conneau et al., 2022) and the **Mann-ki-Baat**, a subset of the BhasaAnuvaad collection[2] (Sankar et al., 2025a). Since the datasets differ in structure and annotation format, different preprocessing strategies are applied before training and evaluation.

FLEURS (Conneau et al., 2022) is a multilingual speech dataset containing speech recordings and aligned transcriptions across more than 100 languages. We use the English and Dravidian language subsets (Tamil, Telugu, and Kannada). Since the dataset is n-way parallel, samples across languages share the same identifier; we align English and target language sentences using their unique IDs and remove samples with missing or empty transcriptions. The predefined training and test splits are used directly: training for fine-tuning and test for both zero-shot and fine-tuned evaluation. Audio inputs are decoded, converted to mono when required, and resampled to 16 kHz before ASR processing.

The Mann-ki-Baat dataset contains speech recordings from the Mann-ki-Baat radio program with English transcripts and translations in multiple Indian languages, but does not provide explicit Dravidian-to-Dravidian pairs. We construct parallel data via automatic alignment using `LaBSE` sentence embeddings (`sentence-transformers/LaBSE`) (Reimers and Gurevych, 2019). Cosine similarity scores are computed between sentence pairs, followed by Hungarian one-to-one matching. Only pairs exceeding a cosine similarity threshold of 0.6 are retained, yielding approximately 15,284 aligned pairs and enabling the construction of semantically consistent pseudo-parallel Dravidian language pairs. The dataset is shuffled with a fixed random seed (42) and split into train/development/test sets using an 80/10/10 ratio, with identical splits reused across both systems for fair comparison. For English-to-Dravidian translation, source audio is directly used for ASR transcription, while samples with missing audio, empty transcripts, or target translations are discarded before splitting. Audio segments are decoded, converted to mono if necessary, and resampled to 16 kHz to ensure consistent input format across both systems, as Whisper, IndicConformer, and SeamlessM4T-v2 all require fixed-rate mono audio for stable training and inference.

[2] https://huggingface.co/datasets/ai4bharat/Mann-ki-Baat

FLAICOL: Flip-Point-Led Augmentation for Imbalanced Code-Mixed Offensive Language Detection

Danish Mohammed and Vidhya Kamakshi
National Institute of Technology Calicut, Kozhikode - 673601, Kerala, India.
{danish_m240681cs,vidhyakamakshi}@nitc.ac.in

Abstract

Hate speech detection in low-resource, code-mixed languages is a challenging task as people often switch between scripts and languages in a single post. Code-Mixed scripts can take the form of explicit slurs, subtle insults, or fragmented abuse, and is often hidden by spelling variants and Romanized script. These datasets are also subjected to class imbalance with hate speech being a minority class of interest. To mitigate the imbalance, targeted data augmentation of minority class samples can help learn better representations to aid hate speech detection despite the naturally expected imbalance. We propose FLAICOL, a flip-point method which identifies the minimal embedding perturbation that moves an input across the decision boundary, map it back to discrete text, and retrain on those focused examples. Empirical results show that these interpretable augmentations strengthen Transformer classifiers on low-resource, code-mixed low resource hate datasets (Experiments were conducted on the Tamil-English, Malayalam-English, and Kannada-English splits in the Dravidian CodeMix Benchmark).

1 Introduction

Hate speech on social media has grown sharply in recent years, with platforms reporting a steady rise in abusive and offensive content. In multilingual regions such as India, users often mix native scripts (e.g., Tamil, Malayalam) with Latin tokens (Chakravarthi et al., 2022) in the same message. This mixing leads to lexical variation alternate spellings like "school" vs. "skool" and script alternation that monolingual detectors fail to handle. Simple keyword filters and classical machine-learning models struggle when words appear in unexpected forms or scripts. Early works applied SVMs, Naïve Bayes, and decision-tree classifiers (Das et al., 2023) to code-mixed text but yielded low accuracy because these models cannot learn subword patterns or contextual cues across scripts. Character-level CNNs (Rani et al., 2020) later proved effective at modeling noisy spellings, boosting F1 score on mixed datasets. Transliteration schemes (Roy and Kumar, 2025) that map all text to a single script reduced vocabulary size and improved recall on code-mixed data. Adversarial perturbations in transformer embeddings (Liu et al., 2020) further increased robustness to orthographic noise, showing gains in F1 scores on benchmark splits. Lexicon-injected transformer methods (Sariyanto et al., 2025) better handle rare offensive terms, yielding recall improvements. Cohen et al. (Cohen et al., 2023) leverage back-translation and LLM capabilities for data augmentation to enhance hate speech detection accuracy. This is a general augmentation strategy, unlike our proposed framework, FLAICOL, which performs boundary-aware augmentation.

While these approaches improve robustness, they often lack interpretability (Sharma et al., 2021; Kamakshi and Krishnan, 2023). To improve both performance and transparency, we propose FLAICOL, a framework leveraging a flip-point (a neighboring instance that is close to an instance of interest but flips the prediction) method (Yousefzadeh and O'Leary, 2020) that adapts the closest-flip-point and homotopy framework for pretrained Transformer classifiers. We compute a homotopy set and trace the homotopy back to the original classifier while optimizing small token-embedding perturbations and converting validated continuous solutions into discrete, targeted augmentation examples. To mitigate data scarcity and class imbalance (Thandil et al., 2025), these flip-derived examples are used to enrich offensive samples before fine-tuning. We evaluate our approach on Tamil–English, Malayalam–English, and Kannada-English corpora from the DravidianCodeMix benchmark (Chakravarthi et al., 2022), focusing on improving performance

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 21–31
July 3, 2026 ©2026 Association for Computational Linguistics

for low-resource, code-mixed hate-speech detection. Our results and analysis aim to support the development of more reliable and fair moderation tools and to help create safer online spaces for multilingual communities.

2 Related Work

Early efforts (Das et al., 2023) in hate speech detection relied heavily on traditional machine learning classifiers such as SVMs, Naïve Bayes, and decision trees. These methods struggled to capture contextual nuances and cross-lingual dependencies inherent in code-mixed text. Rani et al. (Rani et al., 2020) demonstrated that character-level CNNs outperformed these classical models by modeling subword patterns, achieving significantly higher F1 scores. Similarly, Saumya et al. (Saumya et al., 2021) showed that in very low-resource settings, character-level n-gram TF–IDF features combined with Naïve Bayes or logistic regression outperformed deep models like BERT (Devlin et al., 2019) and ULM-FiT (Joseph and Joshi, 2024). Roy & Kumar (Roy and Kumar, 2025) further reported that transliterating Telugu–English code-mixed text into a unified script before modeling improved accuracy to 75%, reducing vocabulary variation.

The introduction of adversarial techniques marked an important advance in strengthening multilingual Transformer models. Liu et al. (Liu et al., 2020) fine-tuned ERNIE 2.0 on Hindi–English data and injected adversarial perturbations into XLM-R embeddings, producing F1 gains of up to 5% across ensemble folds. Their results show that small, targeted perturbations in embedding space can act as an effective regularizer. Like adversarial techniques, lexicon-driven methods improve interpretability by tying model decisions to curated word lists and linguistically informed features.

Lexicon-driven methods brought interpretability into hate speech detection. Sariyanto et al. (Sariyanto et al., 2025) introduced the VAD-Baseline method, which maps words to valence, arousal, and dominance scores, using gated summation to identify hateful content. Pamungkas & Patti. (Pamungkas and Patti, 2019) integrated LSTM-based sequence models with HurtLex features, improving recall across languages but experiencing significant performance drops in cross-domain testing. Recent hybrid models (Al Nahian et al., 2025) injected lexicon-derived vectors into transformer architectures like mBERT (Devlin et al., 2019), MuRIL (Khanuja et al., 2021), and XLM-R (Conneau et al., 2020), using concatenation or Bi-LSTM layers, achieving good macro-F1 scores on Hindi–English and on Tamil and Malayalam.

The rise of large pretrained transformers significantly improved code-mixed hate speech detection. Varma et al. (Varma et al., 2022) enhanced Malayalam–English detection by pairing BERT embeddings with a logistic regression head. Chakravarthi et al. (Chakravarthi et al., 2022) demonstrated that XLM-RoBERTa consistently outperforms other multilingual encoders on the DravidianCodeMix dataset. Kumar et al. (Kumar et al., 2020) investigated multilingual joint fine-tuning of mBERT (Devlin et al., 2019), mDistilBERT (Sanh, 2019), and XLM-R (Conneau et al., 2020) for hate speech, sentiment, and event detection, finding moderate gains in noisy conditions.

Cross-lingual transfer and ensembling have further boosted performance across several studies. Ghosh & Senapati (Ghosh and Senapati, 2022) evaluated monolingual and multilingual models on Indic languages, reporting MuRIL's (Khanuja et al., 2021) weighted F1 scores above 0.90 for Bengali and around 0.83 for Hindi. Kakati & Dandotiya (Kakati and Dandotiya, 2024) combined a DConv-BLSTM with MuRIL (Khanuja et al., 2021) for Hinglish (Hindi-English), Tamil–English, and Malayalam-English, achieving F1 scores above 0.94. A few of the recent works (Goswami et al., 2023; Raihan et al., 2023, 2024) consider a realistic code-mixing scenario in which more than two languages are mixed in a single instance, and the authors assess the performance of various transformer architectures on such data.

The availability of targeted datasets has also enabled more focused studies: Anbukkarasi & Varadhaganapathy (Anbukkarasi and Varadhaganapathy, 2023) took a Tamil–English corpus and reported an F1 score of 0.81 in Bi-LSTM, while Kumar et al. (Kumar et al., 2025) introduced LexiLogic, a multimodal system that uses fine-tuned BERT variants with back-translation and synonym replacement to obtain high macro-F1 scores. Kavatagi et al. (Kavatagi and Rachh, 2025) developed HASTIKA, a fine-grained Kannada-English hate and target identification system that achieved an F1 score of around 0.78.

Despite these advances, a major bottleneck that remains is the scarcity of large, diverse annotated corpora for code-mixed hate speech, especially for low-resource languages. The Dravidi-

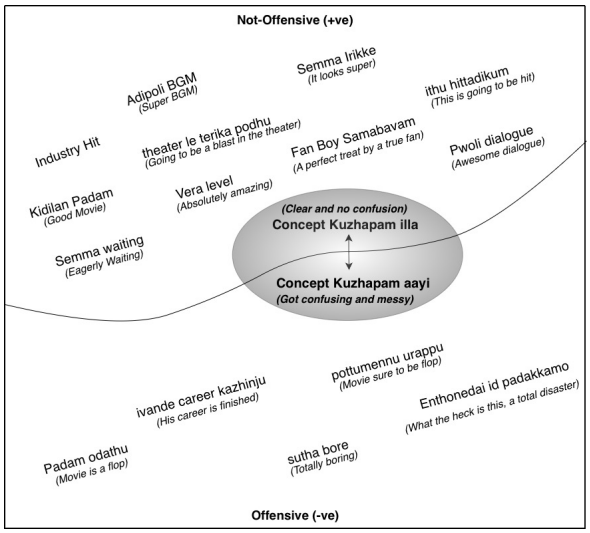


Figure 1: Conceptual embedding-space illustration of FLAICOL, the proposed homotopy flip-point idea. The decision boundary separates Not-Offensive (positive) and Offensive (negative) regions, the shaded oval indicates the homotopy search region and the arrow denotes the minimal semantics-preserving perturbation $\delta_{\min}$ that produces a flip point. The translation of the code-mixed instances is provided in brackets to facilitate global researchers.

anCodeMix dataset introduced by Chakravarthi et al. (Chakravarthi et al., 2022) provides manually labeled code-mixed examples and is a valuable training resource, but its severe class imbalance (Thandil et al., 2025) continues to limit model robustness and generalization. Our proposed methodology, FLAICOL, aims to address the imbalance by augmenting the minority (offensive) class using flip points.

3 Proposed Methodology

The proposed approach FLAICOL is inspired by the flip-point and homotopy framework (Yousefzadeh and O'Leary, 2020), adapted for pretrained Transformer classifiers and discrete text by using a bias homotopy on the classifier head and searching in token embedding space.

As illustrated in Figure 1, our approach searches for minimal semantic perturbations that move an example across the classifier decision boundary. These flip points are decoded into discrete token edits and used for data augmentation.

3.1 FLAICOL, a Flip-point framework

A *flip point* is a minimally perturbed input that lies on the decision boundary between two classes. Formally, for an input x predicted as class i and a chosen target class j, the *closest flip point* $x^\star$ is defined as the solution to the constrained optimization

$$\begin{aligned} x^\star = \arg\min_{x'} \|x' - x\|_2^2 \\ \text{s.t.} \quad z_i(x') = z_j(x') \\ z_i(x') \geq z_k(x') \quad \forall k \notin \{i, j\} \\ x' \in [L, U] \end{aligned} \tag{1}$$

where $z(\cdot)$ denotes the model logits (pre-softmax) and $[L, U]$ denotes feasible input bounds.

Because textual inputs are discrete, we adapt the homotopy idea for Transformers used for binary classification by first computing a minimal change to the classifier bias that makes the given example feasible. Given an encoder representation $h(x)$ and the classifier head parameters (W, b), the homotopy set is obtained by solving

$$\begin{aligned} b_{\text{homo}} = \arg\min_{b'} \|b' - b_{\text{orig}}\|_2^2 \\ \text{s.t.} \quad (h(x)W^\top + b')_i \leq (h(x)W^\top + b')_j \end{aligned} \tag{2}$$

Once a feasible bias b_{homo} is found, we trace (interpolate) the bias back to the original bias b_{orig} and re-solve for a nearby flip at each interpolation step.

To make the constrained search tractable for tokenized text we perform optimization in continuous token-embedding space. Let E_0 denote the original sequence of token embeddings for x, and let Δ denote small continuous offsets applied to a selected subset of token positions. As a practical surrogate for the constrained problem (equation (1)), we optimize the following penalty objective:

$$\mathcal{L}(\Delta) = \|\Delta\|_2^2 + \alpha(z_i(E_0 + \Delta) - z_j(E_0 + \Delta))^2 \tag{3}$$

where $\alpha > 0$ is a penalty weight controlling the enforcement of minimal perturbation to the instance under consideration and flipping of the classification decision. After optimizing (3) , the perturbed embedding $E_0 + \Delta$ is treated as a flip. The continuous solution is then mapped back to discrete tokens by nearest-neighbor lookup in the model's embedding matrix. The decision obtained on the mapped augmented instance is then validated on the original classifier (to ensure that minority-class representative examples are increased through the flip-point-led augmentation).

4 Experimental Setup

Experiments were conducted using Google Colab with GPU acceleration (NVIDIA T4, 16GB VRAM). All models were implemented in PyTorch using the HuggingFace transformers framework. Mixed-precision (FP16) training was enabled when CUDA was available.

We evaluate two pretrained multilingual Transformer backbones, namely XLM-RoBERTa-base (Kodali et al., 2025) and MuRIL-base (Khanuja et al., 2021).

Both models were fine-tuned for binary sequence classification (Not-Offensive vs Offensive).

4.1 Flip-Point Configuration

The following configuration was adopted for both backbones:

- Maximum Sequence Length = 64
- α = {1, 10, 100}
- Learning rate = 10^{-2}
- Number of Homotopy Iterations = 6
- Batch Size = 64
- Optimizer: Adam (Kingma and Ba, 2015)
- Random seed: {1, 2, 3, 42 }.

The implementation can be accessed from `https://github.com/FLAICOL/FLAICOL.git`

4.2 Dataset

We utilize the DravidianCodeMix corpora, which are specifically curated for hate speech detection in code-mixed text. The datasets are prepared with two label categories: "Not-Offensive" and "Offensive". The data consists of user-generated comments from YouTube, reflecting real-world scenarios where Dravidian languages are mixed with English (code-mixing). The details of the specific splits that we use are given below:

- *Tamil–English:* This corpus consists of 42,133 examples. It shows a notable class imbalance, with 31,808 "Not-Offensive" instances and 10,325 "Offensive" instances.
- *Malayalam–English:* This corpus consists of 18,403 examples. It exhibits a severe class imbalance, with 17,697 "Not-Offensive" examples and only 706 "Offensive" examples. This significant skew towards the majority class presents a key challenge for model training.
- *Kannada–English:* This corpus consists of 5874 examples. It shows a notable class imbalance, with 4397 "Not-Offensive" examples and only 1477 "Offensive" examples.

4.3 Implementation

The practical pipeline for implementation of FLAICOL follows three main phases: Fine-tuning, Finding Flip points, Augmentation, and retraining.

4.3.1 Fine-tune base classifier

For each corpus we fine-tune a pretrained Transformer sequence-classification model on the training split. The fine-tuned model provides logits $z(\cdot)$ and encoder outputs $h(\cdot)$ required by the flip discovery procedure.

4.3.2 Find flip points

For each example x chosen for analysis:

1. Compute the model prediction $i = \arg\max z(x)$ and select a target class j (typically the highest-scoring alternative).
2. Extract the encoder representation $h(x)$ and solve the bias subproblem (equation (2)) to obtain b_{homo}.
3. Interpolate the bias from b_{homo} back to b_{orig} in several homotopy steps. At each step fix the temporary bias and perform a local embedding-space search:
   - Select a small set of token positions to perturb.
   - Optimize offsets Δ for those positions by minimizing the penalty objective (3) with a gradient-based optimizer (Adam) and a schedule of penalty weights α.
   - If the equality $z_i \approx z_j$ and the inequality conditions hold, treat the solution as successful and warm-start the next homotopy step with the found offsets.
4. After the final homotopy step, map the continuous perturbed embeddings $E_0 + \Delta$ to discrete tokens via nearest-neighbor in the embedding matrix, run the discrete candidate through the original classifier, and keep validated flips that meet the flip criteria.

4.3.3 Augmentation and retraining

Obtained flip texts are converted into synthetic training examples and appended to the original training data. The classifier is retrained on this augmented dataset, and held-out evaluation is used to measure the effect of flip-based augmentation.

As a simple baseline for minority-class augmentation, we also ran SMOTE (synthetic minority oversampling) (Chawla et al., 2002) on the training split and retrained the same classifier architecture for direct comparison with flip-point augmentation, the backbone behind the proposed FLAICOL.

5 Results

We evaluate FLAICOL using two encoder backbones—XLM-RoBERTa-base (Kodali et al., 2025) and MURIL-base (Khanuja et al., 2021) fine-tuned on the Malayalam-English, Tamil-English, and Kannada-English splits of the Dravidian-CodeMix corpus. For each language we generate a certain number ({100, 150, 200 }) of flip-derived examples and append them to the training set. Overall, FLAICOL yields modest but directionally consistent gains in validation performance, with the largest improvements observed for the minority (Offensive) class, suggesting reduced near-boundary errors.

5.1 Distributional plausibility and surface fluency of flips

We assess reconstructions with three complementary, deterministic diagnostics: (i) lexical/phrase diversity measured by Self-BLEU (S-BLEU$_{2/3/4}$, lower = more diverse), (ii) per-pair semantic fidelity measured by pairwise LaBSE cosine (reported as LaBSE$_{\text{mean}}$ and the 90th percentile LaBSE$_{p90}$) and (iii) set-level distributional similarity measured by a Fréchet Embedding Distance (FED) (Alihosseini et al., 2019) computed on LaBSE embeddings (bootstrap mean with 95% CI). Table 1 reports these statistics for the four experimental conditions.

The table shows a clear backbone-level separation. XLM-R reconstructions preserve semantics substantially better than MuRIL, with LaBSE$_{\text{mean}}$ similarity in the mid 0.7 range and upper-tail (p90) values approaching 0.9, whereas MuRIL remains in the mid 0.3 range. At the set level, distributional drift is also consistently smaller for XLM-R (FED $\approx$ 0.43–0.46) than for MuRIL ($\approx$ 0.62–0.63), and the bootstrap confidence intervals do not overlap, suggesting a statistically stable separation across conditions.

Self-BLEU reveals a complementary trade-off. MuRIL reconstructions show much lower Self-BLEU values (around 0.03–0.06), indicating greater lexical diversity, while XLM-R reconstructions fall in a higher range (roughly 0.08–0.28), reflecting greater phrase reuse. Taken together with the LaBSE and FED results, this suggests that MuRIL achieves higher surface diversity at the cost of semantic fidelity and increased distributional shift, whereas XLM-R maintains closer semantic and distributional alignment with slightly lower lexical variation.

These metrics indicate a consistent encoder-level separation in semantic fidelity and distributional alignment.XLM-R reconstructions remain closer to the original semantic manifold, whereas MuRIL reconstructions exhibit greater stylistic and distributional divergence. This difference reflects how encoder representations shape the stability of flip-based generation.

5.2 Qualitative comparison of flip-finding backbones

Representative reconstructions are shown adjacent to their descriptions to emphasize the surface phenomena that underlie the metrics.

5.2.1 XLM-R reconstructions:

XLM reconstructions typically:

- preserve Tamil/Kannada/Malayalam–English code-mixing and script fidelity,
- retain colloquial, comment-style phrasing,
- introduce small, interpretable lexical and spacing perturbations, and
- maintain semantic similarity to the original comments.

In several cases, the reconstructed text is nearly identical to the original, differing only by additional whitespace, underscores, or minor tokenization artifacts. These minimal surface variations explain the lower FED and higher LaBSE observed for XLM flips, decoded token sequences often remain very close to the originals, as reflected by low normalized token-edit distance and relatively higher Self-BLEU, indicating local and controlled edits rather than substantial rewrites.

Table 1: Evaluation of generated flip-points.

Backbone	Language	S-BLEU$_{2/3/4}$	LaBSE$_{mean}$	LaBSE$_{p90}$	FED$_{mean}$ (95% CI)
XLM-R	Mal	0.285 / 0.139 / 0.083	0.726	0.876	0.461 (0.412–0.512)
XLM-R	Tam	0.271 / 0.108 / 0.058	0.759	0.896	0.436 (0.391–0.483)
XLM-R	Kan	0.249 / 0.159 / 0.121	0.762	0.908	0.433 (0.390–0.481)
MuRIL	Mal	0.058 / 0.041 / 0.035	0.360	0.633	0.631 (0.589–0.687)
MuRIL	Tam	0.060 / 0.037 / 0.030	0.373	0.663	0.618 (0.579–0.659)
MuRIL	Kan	0.122 / 0.081 / 0.064	0.384	0.678	0.602 (0.559–0.635)

language	text	reconstructed	translation
Malayalam	കിടുക്കാച്ചി പടം ആയിരിക്കട്ടെ എല്ലാവിധ ആശംസകളും നേരുന്നു	കി ടുക്ക ാ ച്ചി പട ം ആയിര ിക്ക ട്ടെ എല്ലാ വിധ ആശംസ കളും നേര ുന്നു	May it be an awesome movie. All the best.
Malayalam	Mammokka nigalude hardworkinu nalla vijayam thane kittum	Mam mo kka ni gal ude hard work inu na lla vijay am than e kit tum	Mammokka, your hard work will definitely bring great success.
Tamil	எல்லாம் ok.. இந்த உதயநிதி பய தான் சொதபிடுவானோனு பயம்மா இருக்கு.	எல்லாம் ok இந்த உத ய நிதி பய தான் சொ த பி டு வா னோ ந ா பய ம்மா இருக்கு	Everything is good, but I'm scared Udhayanidhi might mess it up
Tamil	NerKondaParvai Thala fans like panna vanam... தல வெறியன்கள் like பண்ணுங்க	N er Kon da Par vai Thala fans like panna van am த ல வெறி யன் கள் like பண்ண ுங்க	Nerkonda Paarvai Thala fans, show your support... Die-hard Thala fans, hit like
Kannada	ಭಾರತದಲ್ಲಿ ಕಂಪನಿ ಸುರು ಮಾಡಲು ಬಂದ್ರೆ ಸಿಟಿ ಬಿಟ್ಟು ಹಳ್ಳಿಯ ಕಡೆಗೆ ಬರೊದೆ ಇಲ್ಲ.ಇಉತ್ತಮ ಮತ್ತು ವಿಶಾಲ ಪ್ರದೇಶ ಇರೊದು ಹಳ್ಳಿಯಲ್ಲಿ .ಮೊದಲು ಅದನ್ನ ಸರ್ಕಾರ ಮಾಡಬೇಕು.	ಭಾರತದಲ್ಲಿ ಕಂಪನಿ ಸು ರು ಮಾಡಲು ಬಂದ್ ರೆ ಸಿ ಟಿ ಬಿಟ್ಟು ಹಳ್ಳಿ ಯ ಕಡೆ ಗೆ ಬರ ೊ ದೆ ಇಲ್ಲ ಇ ಉ ತ್ತ ಮ ಮತ್ತು ವಿ ಶಾ ಲ ಪ್ರದೇಶ ಇರ ೊ ದು ಹಳ್ಳಿ ಯಲ್ಲಿ ಮ ೊ ದ ಲು ಅದ ನ್ನ ಸರ್ಕಾರ ಮಾಡಬೇಕು	When companies come to start in India, they don't look beyond cities. The best and most spacious areas are in villages; the government should prioritize that first.
Kannada	ತುಂಬಾ ಚನ್ನಾಗಿ ಮೂಡಿ ಬಂದಿದೆ ಮತ್ತು ನಿರೀಕ್ಷೆ ಹುಟ್ಟಿಸುವಂತಿದೆ. ಸಿನಿಮಾ ಬಿಡುಗಡೆಗೆ ಕಾಯುತ್ತೇವೆ. ಒಳ್ಳೆಯದಾಗಲಿ	ತುಂಬಾ ಚ ನ್ನ ಾಗಿ ಮೂ ಡಿ ಬಂದಿದೆ ಮತ್ತು ನಿರೀಕ್ಷೆ ಹುಟ್ಟ ಿಸುವ ಂತ ಿದೆ ಸಿನಿಮಾ ಬಿಡುಗಡೆ ಗೆ ಕಾಯ ುತ್ತೇವೆ ಒಳ್ಳೆಯ ದಾಗ ಲಿ	It has come out very well and looks promising. Waiting for the release; all the best!

Figure 2: Examples of XLM-based flip reconstructions.

5.2.2 MuRIL reconstructions:

MuRIL reconstructions frequently:

- include cross-script insertions (e.g., Odia, Bengali, Devanagari fragments),
- contain multi-language segments unrelated to the input context,
- show large structural edits and irregular token boundaries and
- display pronounced semantic drift from the source comment.

These behaviors explain the higher FED and lower LaBSE for MuRIL flips, decoded sequences exhibit larger surface changes and greater lexical novelty, reflected in higher normalized token-edit distance and lower Self-BLEU.

Lower FED together with higher LaBSE similarity and small normalized token-edit distance indicates that XLM reconstructions remain close to the source distribution while preserving semantic structure and surface patterns. In contrast, higher FED combined with lower LaBSE, larger normalized edits, and lower Self-BLEU for MuRIL reconstructions reflects stronger stylistic and distributional divergence, often including noisier or cross-script variations. Practically, these findings suggest that XLM flips are more directly aligned with the original data manifold, whereas MuRIL flips require stricter similarity or structural constraints prior to inclusion.

Table 2: PMI-based analysis for the offensive class across MURIL and XLM settings.

Dataset	Original offensive PMI	Flip offensive PMI
MAL-XLM	4.79%	0.00%
TAM-XLM	0.71%	0.00%
KAN-XLM	2.70%	0.55%
MAL-MuRIL	4.79%	0.00%
TAM-MuRIL	0.71%	0.29%
KAN-MuRIL	2.70%	0.00%

5.2.3 PMI analysis

As shown in Table 2, flip-generated offensive samples exhibit substantially lower overlap with high-

language	text	reconstructed	translation
Malayalam	Apt മ്യൂസിക് Sushin.. Hats off	Apt ഢ്വിഓ± ധ.ഖിഓ५ലേഖാംച്ചി 1797 ഫ്ലെയ്ക്കുs ഥ്വാഥ ഗ്രീമാർി	Spot on music, Sushin... Hats off, spot on
Malayalam	രാക്ഷസൻ ഇമൈകൾനോടികൾ ഒക്കെ അങ്ങോട്ട് മാറിനിക്ക് ഇനി ഇവിടെ 5-ആം പാതിര ഭരിക്കും	രാക്ഷസൻ ഇമൈകൾനോടികൾ 6500 അ ടാകനാമനിക്ക് ക്രിടാംഗണ ഇവിടെ 5 ആം TDുഷാ ഭ പഞ്ചാങ്ഗാനുസാരं जवळपासची	Step back, Ratsasan and Imaikkaa Nodigal. Anjaam Pathira is the new ruler of this genre
Tamil	BGM asusual therikka vitaru Sam cs	BGM కొత్తవీధి وزير खेलकुदकोu نياپو cs 139	As usual, Sam CS is just blowing it up with the BGM
Tamil	Waiting for lokesh kanakaraj Thalapathy 64 Thalapathy makkal selvam 2020 summer boxoffice	for lokesh kanakaraj Thalap निकिदा। 64 Thalap मकैको वहाँकोा 600px 2020 summer box 380ice,Waiting for Lokesh Kanagaraj's Thalapathy 64, "Thalapathy and Makkal Selvam are going to own the 2020 Summer box office"	Waiting for Lokesh Kanagaraj's Thalapathy 64. Thalapathy and Makkal Selvam are going to own the 2020 Summer box office
Kannada	ಅದ್ಭುತವಾಗಿದೆ ಅವನೇ ಶ್ರೀಮನ್ನಾರಾಯಣ ಚಿತ್ರದ ಹಾಡು ಹ್ಯಾಂಡ್ಸಪ್	അക്ഷരമാലയിലെದ്ബു थानि गीगीर श्रीमन्नारायण బెన్నిన వారీడాసెర కైతప్రం ഘഥിക	Amazing song, 'Hands Up' from Avane Srimannarayana.
Kannada	ಸೂಪರ್ ಸಾಂಗ್ ರಕ್ಷಿತ್ ಶೆಟ್ಟಿ ಸರ್ ನಮ್ಮ ಉಡುಪಿ ಜಿಲ್ಲೆಯ ಹೆಮ್ಮೆ	సూపర్ ہکیزہ రక్షిత్ నేపాలమే బిహాగ సీమాత శబ్దాచీ ਮ।ਗਿਥ। ਤਪज़िले	Great song! Rakshit Shetty sir is the pride of our Udupi.

Figure 3: Examples of MuRIL-based flip reconstructions.

PMI offensive tokens than the original offensive data across both XLM-R and MuRIL settings. Most configurations show zero overlap, with only minor residual overlap observed for TAM-MuRIL and KAN-XLM. These results suggest that the generated flips are not primarily driven by dominant offensive lexical triggers, indicating that FLAICOL does not merely exploit simple PMI-based shortcuts.

5.2.4 Acceptance Rate

To quantify how often a latent-space flip remains valid after discretization, we measure the acceptance rate, defined as the proportion of perturbations that still produce a label change after projection back to token space. This provides a practical measure of discrete survivability, since a perturbation is only useful for augmentation if the label flip persists after continuous-to-discrete conversion.

MuRIL-based settings achieve acceptance rates of 44% (MAL-MuRIL), 53% (TAM-MuRIL), and 71% (KAN-MuRIL), indicating that a substantial fraction of latent flips survive discretization. In contrast, XLM-R settings produce very few perturbations that satisfy the strict post-projection flip validation criterion, despite frequent success during latent-space optimization. Qualitative inspection shows that many XLM-R reconstructions remain extremely close to the original inputs, often differing only in whitespace or tokenization artifacts. This suggests that the model may already be relatively invariant to such minor surface perturbations, causing many projected candidates to remain within the original decision region after discretization.

5.3 Classification impact of flip-point augmentation

We included 100 flip-based augmentation samples, along with larger augmentation settings of 150 and 200 samples, and evaluated all configurations across three random seeds. The resulting Offensive-F1 scores are summarized in Table 3.

As shown in Table 3, FLAICOL consistently improves Offensive-F1 across all six language–model configurations. For XLM-R, the largest gains are generally observed at 200 augmentation samples, with Malayalam showing the strongest improvement (+3.89% points over baseline). For MuRIL, the best performance is typically obtained at 150 samples, particularly for Kannada (+2.84% points), suggesting that moderate augmentation levels may provide a better balance between diversity and stability. To assess robustness, we performed paired t-tests between the baseline and the best-performing augmentation setting for each configuration (last column of Table 3). Although improvements were consistently observed across seeds, statistical significance was not reached under the current setup, likely due to the modest gains.

Table 4 compares flip-point augmentation with a standard SMOTE baseline (Chawla et al., 2002) (both methods appended 100 synthetic samples). Overall, flip-point consistently matches or slightly outperforms SMOTE , where flip-derived examples produce a clearer uplift in minority-class F1.

Table 3: Offensive-class F1 scores (mean ± standard deviation over 3 random seeds) for augmentation settings with 100, 150, and 200 flip-derived samples. $p^{\dagger}$ denotes the paired t-test p-value between the baseline and the best-performing augmentation setting for each language–model configuration.

Language	**Model**	**Offensive-F1 (Mean ± Std)**				$\mathbf{p}^{\dagger}$
		Baseline	**+100**	**+150**	**+200**	
Malayalam	XLM-R	0.6467 ± 0.0459	0.6534 ± 0.0245	0.6569 ± 0.0427	**0.6856 ± 0.0217**	0.116
Tamil	XLM-R	0.6669 ± 0.0036	0.6684 ± 0.0062	0.6754 ± 0.0078	**0.6830 ± 0.0070**	0.118
Kannada	XLM-R	0.6215 ± 0.0159	0.6233 ± 0.0124	0.6268 ± 0.0183	**0.6374 ± 0.0078**	0.259
Malayalam	MuRIL	0.6477 ± 0.0181	0.6563 ± 0.0542	0.6612 ± 0.0103	**0.6635 ± 0.0215**	0.104
Tamil	MuRIL	0.6831 ± 0.0016	0.6872 ± 0.0173	**0.6881 ± 0.0099**	0.6868 ± 0.0089	0.148
Kannada	MuRIL	0.6435 ± 0.0102	0.6505 ± 0.0061	**0.6719 ± 0.0193**	0.6712 ± 0.0292	0.158

Table 4: Validation F1 (Offensive) — comparison of SMOTE vs FLAICOL

Model	Language	SMOTE	FLAICOL (Proposed)
XLM-R	Malayalam	0.626	**0.641**
XLM-R	Tamil	0.654	**0.664**
XLM-R	Kannada	0.611	**0.615**
MURIL	Malayalam	0.704	**0.726**
MURIL	Tamil	0.669	**0.671**
MURIL	Kannada	0.626	**0.640**

Compared to the non-augmented baseline, only a subset of augmentation runs produce gains: some SMOTE configurations did not outperform the base model, whereas flip-point more reliably matched or exceeded baseline performance. This pattern suggests (i) flip samples are most helpful when they introduce meaning-preserving, targeted variations near decision boundaries, and (ii) when generated samples are largely redundant with existing data their incremental benefit is smaller.

XLM flips produce reconstructions that are distributionally closer to the original instances and more semantically faithful, aligning well with the original data manifold. MuRIL flips exhibit greater stylistic variation and distributional shifts and therefore may benefit from stronger similarity or structural constraints before inclusion. Although absolute metric values depend on embedding choice and preprocessing configuration, the relative separation between XLM and MuRIL remains consistent across diagnostics.

6 Summary

Hate-speech detection in code-mixed social media text remains challenging due to script variation, language mixing, and scarce annotated data. We developed FLAICOL, a practical flip-point pipeline for pretrained Transformer classifiers that traces a homotopy-based path to produce small, targeted token-level perturbations and converts continuous solutions into discrete augmentation examples. Applied to the Malayalam–English, Tamil–English and Kannada-English splits of DravidianCodeMix, appending a small set of flips yields measurable robustness gains; overall validation performance improves (macro-F1 increases by roughly one percentage point on our primary runs), and minority-class performance improves modestly.

Beyond these overall gains, our analysis reveals two consistent findings. Reconstructions produced by the XLM encoder remain distributionally and semantically closer to the original corpus, as indicated by lower Fréchet Embedding distance and higher embedding-based similarity, remaining closer to the original embedding distribution. MuRIL reconstructions exhibit greater stylistic divergence and larger distributional shift, reflected in higher Fréchet Embedding distance, lower semantic similarity, and increased lexical variation. These patterns indicate that the choice of encoder meaningfully shapes the semantic stability and distributional realism of flip-generated samples.

FLAICOL combines interpretability with data efficiency; a small set of boundary-targeted flips can strengthen near-decision examples and yield modest but consistent improvements in low-resource, script-diverse, code-mixed settings.

7 Practical implications and Limitations

XLM-style flip generation yields reconstructions that are distributionally closer, more fluent, and more reliable for augmentation in code-mixed Tamil, Kannada and Malayalam data. MuRIL-style reconstructions, while capable of yielding

classification gains, introduce substantial stylistic noise and therefore benefit from stronger filtering (e.g., script constraints or stricter similarity thresholds). Absolute metric values depend on the chosen embedding model (LaBSE) and language-model configuration. However, the relative separation between XLM and MuRIL observed here is consistent across our evaluations. Flip points help us to understand how the model makes decisions about its prediction boundaries. Specifically, if small, non-semantic changes to an input result in a change to the predicted label, it shows that the classifier is dependent on shallow or non-semantic signals and not on robust semantic signals. From this perspective, FLAICOL is a diagnostic tool to show weaknesses in representations that have been learned from the dataset or to indicate that a classifier may rely on artifacts created by the dataset they were trained with or token-level biases. This also fits the larger goal of making models more reliable, particularly in low-resource and noisy code-mixed contexts. Although FLAICOL consistently improves Offensive-F1 across configurations, statistical significance was not reached under the current experimental setup. This is likely due to the modest absolute gains and the limited number of random seeds evaluated. While the experiments in this paper use Dravidian CodeMix benchmark, future work can extend this to other benchmarks. In particular, it would be interesting to examine the Bengali-Hindi-English trilingual code-mixed corpus curated in recent efforts, which, although slightly falling outside the scope of this workshop focusing on NLP techniques in Dravidian languages, shall be of interest to the broader low-resource NLP research community. Future work shall also explore combining flip-based augmentation with generative methods such as back-translation.

Acknowledgements

The authors place their heartfelt gratitude to the resources provided by the institute to which they are affiliated and their department, which is sponsored by the prestigious DST-FIST Initiative of the Government of India. We sincerely thank the reviewers whose constructive feedback has helped shape the camera-ready draft of the paper.

References

Abdullah Al Nahian, Mst Rafia Islam, Azmine Toushik Wasi, and Md Manjurul Ahsan. 2025. Nlpopsciol@dravidianlangtech 2025: Classification of abusive tamil and malayalam text targeting women using pre-trained models. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 38–45.

Danial Alihosseini, Ehsan Montahaei, and Mahdieh Soleymani Baghshah. 2019. Jointly measuring diversity and quality in text generation models. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 90–98, Minneapolis, Minnesota. Association for Computational Linguistics.

S Anbukkarasi and S Varadhaganapathy. 2023. Deep learning-based hate speech detection in code-mixed tamil text. *IETE Journal of Research*, 69(11):7893–7898.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John P McCrae. 2022. Dravidiancodemix: Sentiment analysis and offensive language identification dataset for dravidian languages in code-mixed text. *Language Resources and Evaluation*, 56(3):765–806.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Seffi Cohen, Dan Presil, Or Katz, Ofir Arbili, Shvat Messica, and Lior Rokach. 2023. Enhancing social network hate detection using back translation and gpt-3 augmentations during training and test-time. *Information Fusion*, 99:101887.

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

Subhajeet Das, Koushikk Bhattacharyya, and Sonali Sarkar. 2023. Performance analysis of logistic regression, naive bayes, knn, decision tree, random forest and svm on hate speech detection from twitter. *International Research Journal of Innovations in Engineering and Technology*, 7(3):24.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

Koyel Ghosh and Apurbalal Senapati. 2022. Hate speech detection: a comparison of mono and multilingual transformer model with cross-language evalu-

ation. In *Proceedings of the 36th Pacific Asia Conference on Language, Information and Computation*, pages 853–865.

Dhiman Goswami, Nishat Raihan, Antara Mahmud, Antonios Anastasopoulos, and Marcos Zampieri. 2023. Offmix-3l: A novel code-mixed test dataset in bangla-english-hindi for offensive language identification. In *Proceedings of the 11th International Workshop on Natural Language Processing for Social Media*, pages 21–27.

Shenson Joseph and Herat Joshi. 2024. Ulmfit: universal language model fine-tuning for text classification. *International Journal of Advanced Medical Sciences and Technology*, 4(6):10–54105.

Pallabi Kakati and Devendra Dandotiya. 2024. Automatic detection of hate speech in code-mixed indian languages in twitter social media interaction using dconvblstm-muril ensemble method. *Social Network Analysis and Mining*, 14(1):108.

Vidhya Kamakshi and Narayanan C Krishnan. 2023. Explainable image classification: The journey so far and the road ahead. *AI*, 4(3):620–651.

Sanjana Kavatagi and Rashmi Rachh. 2025. Hastika: hate speech and target identification in kannada-english code-mixed text. *Language Resources and Evaluation*, pages 1–46.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, and 1 others. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Diederik P Kingma and Jimmy L Ba. 2015. Adam : A method for stochastic optimization. *International Conference on Learning Representations*, 7.

Rohith Gowtham Kodali, Durga Prasad Manukonda, and Daniel Iglesias. 2025. bytesizedllm@ nlu of devanagari script languages 2025: Hate speech detection and target identification using customized attention bilstm and xlm-roberta base embeddings. In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025)*, pages 242–247.

Niranjan Kumar, Pranav Gupta, Billodal Roy, and Souvik Bhattacharyya. 2025. Lexilogic@ dravidian-langtech 2025: Detecting misogynistic memes and abusive tamil and malayalam text targeting women on social media. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 435–439.

Ritesh Kumar, Bornini Lahiri, Atul Kr Ojha, and Akanksha Bansal. 2020. Comma@ fire 2020: Exploring multilingual joint training across different classification tasks. In *FIRE (Working Notes)*, pages 823–828.

Jiaxiang Liu, Xuyi Chen, Shikun Feng, Shuohuan Wang, Xuan Ouyang, Yu Sun, Zhengjie Huang, and Weiyue Su. 2020. Kk2018 at SemEval-2020 task 9: Adversarial training for code-mixing sentiment classification. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 817–823, Barcelona (online). International Committee for Computational Linguistics.

Endang Wahyu Pamungkas and Viviana Patti. 2019. Cross-domain and cross-lingual abusive language detection: A hybrid approach with deep learning and a multilingual lexicon. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 363–370.

Md Nishat Raihan, Dhiman Goswami, Antara Mahmud, Antonios Anastasopoulos, and Marcos Zampieri. 2023. Sentmix-3l: A bangla-english-hindi code-mixed dataset for sentiment analysis. In *Proceedings of the First Workshop in South East Asian Language Processing*, pages 79–84.

Nishat Raihan, Dhiman Goswami, Antara Mahmud, Antonios Anastasopoulos, and Marcos Zampieri. 2024. Emomix-3l: a code-mixed dataset for bangla-english-hindi for emotion detection. In *Proceedings of the 7th Workshop on Indian Language Data: Resources and Evaluation*, pages 11–16.

Priya Rani, Shardul Suryawanshi, Koustava Goswami, Bharathi Raja Chakravarthi, Theodorus Fransen, and John Philip McCrae. 2020. A comparative study of different state-of-the-art hate speech detection methods in hindi-english code-mixed data. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 42–48.

Pradeep Kumar Roy and Abhinav Kumar. 2025. Ensuring safety in digital spaces: Detecting code-mixed hate speech in social media posts. *Data & Knowledge Engineering*, page 102409.

V Sanh. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *Proceedings of Thirty-third Conference on Neural Information Processing Systems (NIPS2019)*.

Happy Khairunnisa Sariyanto, Diclehan Ulucan, Oguzhan Ulucan, and Marc Ebner. 2025. Towards explainable hate speech detection. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12883–12893.

Sunil Saumya, Abhinav Kumar, and Jyoti Prakash Singh. 2021. Offensive language identification in dravidian code mixed social media text. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 36–45.

Rajat Sharma, Nikhil Reddy, Vidhya Kamakshi, Narayanan C Krishnan, and Shweta Jain. 2021. Maire-a model-agnostic interpretable rule extraction procedure for explaining classifiers. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 329–349. Springer.

Rizwana Kallooravi Thandil, Muneer Vk, and Sabique Pv. 2025. A novel deep learning framework with advanced feature engineering for hate speech detection in accented malayalam speech. *Humanities and Social Sciences Communications*, 13(1):10.

P Deepasree Varma, P Vinod, M Nandakumar, K Akshay, and Akhil Madhu. 2022. Hate speech detection in english and malayalam code-mixed text using bert embedding. In *2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS)*, pages 1–6.

Roozbeh Yousefzadeh and Dianne P. O'Leary. 2020. Deep learning interpretation: Flip points and homotopy methods. In *Proceedings of The First Mathematical and Scientific Machine Learning Conference*, volume 107 of *Proceedings of Machine Learning Research*, pages 1–26. PMLR.

LIMP: Linguistically-Informed Multi-Strategy Prompting for Telugu Multi-Turn Dialogue Generation

Arjungopal Anilkumar[1], **Suryansh Ram Menon**[1], **Divagar S**[1], **Premjith B**[1]

Amrita School of Artificial Intelligence, Coimbatore, Amrita Vishwa Vidyapeetham, India[1]

Correspondence: b_premjith@cb.amrita.edu

Abstract

Generating contextually coherent multi-turn dialogue in Telugu requires resolving three deeply interacting constraints absent from generic LLM prompting: morphologically encoded social hierarchy (honorific verb conjugations), strict SOV agglutinative syntax, and culturally governed emotional logic formalised in Natyashastra *rasa* theory (Bharata Muni, 1951). We introduce LIMP (**L**inguistically-**I**nformed **M**ulti-Strategy **P**rompting), an inference-time, training-free framework that injects expert linguistic and cultural knowledge into prompt structure, requiring no fine-tuning or labelled data. We empirically evaluate seven configurations spanning four prompting strategies across two model backbones on 500 stratified evaluation instances from the IndicDialogue Telugu corpus (Arnob et al., 2024): a zero-shot baseline, LIMP-RAW (dense constraint prompt), LIMP-COT (six-stage analytical scaffold grounded in rasa theory and Telugu morphological grammar), and a context-compressed CoT variant for SARVAM-1. LIMP-COT achieves 2× higher Jaccard and Dice than LIMP-RAW on GEMMA-3-1B-IT (Gemma Team, Google DeepMind, 2025) (1B parameters): Jaccard = 0.0140 vs. 0.0098, Dice = 0.0274 vs. 0.0193 ($p < 0.001$), demonstrating that sequential analytical commitment to linguistic constraints produces more form-faithful Telugu than holistic constraint injection. Under LLM-as-judge evaluation, LIMP-COT achieves the highest Adequacy (3.90 ± 0.91) among GEMMA-3-1B-IT configurations, while LIMP-RAW achieves the highest Fluency (4.07 ± 0.93). The zero-shot baseline achieves the highest BERTSCORE F_1 (0.9760) across all configurations, revealing a three-way dissociation: automatic semantic metrics favour zero-shot generation, fluency favours dense constraint prompting, and contextual adequacy favours sequential scaffolding. This *semantic–lexical–pragmatic dissociation*, where no single configuration dominates across all metric classes, is itself a substantive finding: in agglutinative Telugu, semantic paraphrase fidelity, morphosyntactic surface fidelity, and pragmatic adequacy are orthogonal evaluation dimensions that must be assessed concurrently.

1 Introduction

Multi-turn dialogue generation is the task of producing contextually coherent responses conditioned on a sequence of prior conversational turns. Unlike single-turn generation, the model must jointly track discourse state, speaker intent, and pragmatic coherence across multiple exchanges while avoiding generic or context-insensitive outputs—a challenge that has driven a research trajectory from hierarchical recurrent architectures (Serban et al., 2016) through large-scale conversational pretraining (Zhang et al., 2020b) to instruction-tuned generative models (Brown et al., 2020). Diversity and specificity in generation remain persistent failure modes (Li et al., 2016; Holtzman et al., 2020), and even frontier LLMs require careful conditioning to maintain coherence across long conversational contexts (Roller et al., 2021; Bang et al., 2023).

Multi-turn dialogue generation in Telugu—a morphologically rich Dravidian language spoken by approximately 82 million people (Eberhard et al., 2023)—poses constraints of a qualitatively different order. Telugu verb conjugations encode the speaker-addressee power relationship as a grammatical obligation: the distinction between honorific endings such as *-tunnaaru* and familiar endings such as *-tunnaavu* is a categorical social obligation (Krishnamurti, 2003; Steever, 1998). Telugu is additionally a canonical SOV language whose agglutinative morphology yields a combinatorially large surface paradigm; models trained predominantly on Indo-European corpora exhibit degraded cross-lingual transfer performance on typologically distant languages (Wu and Dredze, 2019), and systematically produce word-order errors in SOV languages such as Telugu (Subbarao, 2012). A further constraint, unique to the cinematic domain, is *rasa*, the classical Indian theory of aesthetic emotion codified in the Natyashastra (Bharata Muni,

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 32–41
July 3, 2026 ©2026 Association for Computational Linguistics

1951): each conversational turn is governed by one of nine canonical emotional categories (Sreejith et al., 2017). These three constraints operate simultaneously and interdependently, and Dravidian languages remain systematically underserved by both multilingual pretraining corpora and existing dialogue architectures (Joshi et al., 2020; Ahuja et al., 2023).

Contemporary LLMs exhibit broad multilingual competence yet do not encode these Telugu-specific morphological and cultural priors. Generic chain-of-thought (CoT) prompting (Wei et al., 2022) decomposes reasoning into ordered steps but defines those steps task-abstractly, providing no guidance for culturally or morphologically grounded generation (Shi et al., 2023), and low-resource settings preclude the data-intensive fine-tuning paradigm (Lauscher et al., 2020). We introduce LIMP (**L**inguistically-**I**nformed **M**ulti-Strategy **P**rompting), an inference-time, training-free framework that injects expert linguistic and cultural knowledge directly into the prompt, requiring no labelled data or parameter updates. We empirically evaluate seven configurations—four for GEMMA-3-1B-IT (Zero-Shot, No-LIMP, LIMP-RAW, LIMP-COT) and three for SARVAM-1 (Zero-Shot, LIMP-RAW, CoT-Compressed)—on 500 statistically independent evaluation instances from the IndicDialogue Telugu corpus (Arnob et al., 2024) using nine complementary metrics spanning semantic fidelity, lexical overlap, character-level morphology, and generation diversity. LIMP-COT on GEMMA-3-1B-IT (Gemma Team, Google DeepMind, 2025) (1B parameters) achieves 2× higher Jaccard and Dice than LIMP-RAW on the same backbone (Jaccard = 0.0140 vs. 0.0098; Dice = 0.0274 vs. 0.0193; $p < 0.001$), demonstrating that sequential analytical commitment to linguistic constraints produces more form-faithful Telugu than holistic constraint injection. Under LLM-as-judge evaluation, LIMP-COT achieves the highest Adequacy (3.90 ± 0.91) among GEMMA-3-1B-IT configurations, confirming that the CoT scaffold anchors the model more tightly to the pragmatic demands of the dialogue context. A concurrent finding is a *semantic–lexical–pragmatic dissociation*: the zero-shot baseline achieves the highest BERTSCORE F_1 (0.9760), LIMP-RAW achieves the highest Fluency (4.07), and LIMP-COT achieves the highest Adequacy (3.90), confirming that these are orthogonal evaluation dimensions in agglutinative Telugu.

2 Related Work

Neural multi-turn dialogue generation has progressed from hierarchical recurrent encoder-decoders (Serban et al., 2016) through large-scale pretraining on conversational corpora (Zhang et al., 2020b) to modular open-domain systems (Roller et al., 2021). Maximum mutual information objectives (Li et al., 2016) and nucleus sampling (Holtzman et al., 2020) address the generic response problem, while in-context few-shot learning (Brown et al., 2020) enables generative control without parameter updates. Chain-of-thought prompting (Wei et al., 2022) substantially improves LLM reasoning through step-by-step decomposition, and its extensions—zero-shot (Kojima et al., 2022), self-consistent (Wang et al., 2023), tree-structured (Yao et al., 2023), least-to-most (Zhou et al., 2023), and complexity-based (Fu et al., 2023)—establish staged analytical commitment as a robust mechanism for compositionally difficult tasks. All existing CoT formulations, however, define reasoning stages task-abstractly without encoding domain-expert linguistic knowledge, and distracting or erroneous intermediate steps propagate through the reasoning chain and degrade generation quality (Shi et al., 2023). The NusaCrowd initiative (Cahyawijaya et al., 2023) establishes open-source NLP resources and benchmarks for Indonesian languages—the closest regional parallel to Dravidian resource work—but does not address culturally grounded generation or prompting strategies; no prior work has grounded CoT stages in Natyashastra rasa theory or Telugu morphological grammar.

The Indic NLP landscape has been significantly expanded by a sequence of infrastructure contributions: MURIL (Khanuja et al., 2021) provides BERT-scale multilingual pretraining over 17 Indian languages; the IndicNLP Suite (Kakwani et al., 2020) and AI4Bharat corpora (Kunchukuttan et al., 2020) establish monolingual benchmarks; XLM-R (Conneau et al., 2020) demonstrates large-scale cross-lingual transfer; and IndicTrans2 (Gala et al., 2023) sets the frontier for Indic translation. Despite this progress, Dravidian languages remain systematically underrepresented, with persistent performance gaps documented across frontier models (Joshi et al., 2020; Ahuja et al., 2023). BPE tokenisation conflates morphological boundaries in agglutinative languages (Sennrich et al., 2016b; Bostrom and Durrett, 2020; Mielke et al., 2021), and multilingual tokenisers consistently under-

segment Dravidian text (Rust et al., 2021), directly inflating subword overlap metrics for semantically distinct forms. BERTSCORE (Zhang et al., 2020a) is substantially more robust to agglutinative surface variation than BLEU (Papineni et al., 2002), whose failure modes for morphologically rich languages are well-documented (Callison-Burch et al., 2006; Post, 2018; Mathur et al., 2020); these findings collectively motivate our Unicode purity filter and multi-metric evaluation design.

Computational politeness research has proceeded from foundational politeness theory (Brown and Levinson, 1987) through NMT side constraints for T-V distinction control (Sennrich et al., 2016a), polite dialogue generation (Niu and Bansal, 2018), politeness style transfer (Madaan et al., 2020), and comprehensive landscape surveys (Prabhakaran et al., 2024). All of this work assumes binary or gradient politeness scales that are structurally inadequate for Telugu's multi-tier categorical honorific system (informal, polite, formal) (Krishnamurti, 2003; Steever, 1998). Rasa theory has recently been applied computationally to sentiment analysis (Sreejith et al., 2017) and Sanskrit poetry (Sandhan et al., 2023), with calls to move beyond Western-centric emotion taxonomies in NLP (Plaza-del Arco et al., 2024). In all prior work reviewed here, rasa functions as a post-hoc classification label applied to existing text. LIMP-COT instead positions rasa identification as the upstream Stage 1 commitment from which all morphological and pragmatic decisions are derived, treating rasa as a forward constraint on generation rather than a retrospective annotation. Whether this scaffolding produces outputs that are genuinely rasa-coherent remains an open empirical question requiring human evaluation by domain experts (§6).

3 Methodology

Figure 1 illustrates the end-to-end LIMP pipeline. Raw IndicDialogue data is subjected to a Telugu Unicode purity filter, segmented into five-turn evaluation instances by a non-overlapping sliding window, stratified into a 500-sample evaluation corpus, and passed through one of the evaluated LIMP prompting strategies before decoding by the generation backbone. Generated outputs are evaluated against gold references using nine complementary metrics.

We use the Telugu subset of the IndicDialogue corpus, a subtitle-derived dataset of 10 Indic languages distributed as line-delimited JSONL files (Arnob et al., 2024), comprising Telugu-language subtitle data sourced from OpenSubtitles.org (75 files; ≈90,950 dialogue instances). The corpus includes both original Telugu productions and dubbed content; dubbed material may not exhibit authentic rasa arcs or native honorific patterns, a limitation acknowledged in §6. Where the data derives from original Telugu productions, the cinematic domain encodes speaker hierarchy, emotional rasa arc, and regional dialect simultaneously in each turn—making it a substantive testbed for the three generation constraints described in §1. Telugu NLP data is heavily contaminated by transliteration, code-mixing, and subtitle processing artifacts (Kunchukuttan et al., 2020). We therefore apply a strict Unicode purity filter requiring every script-carrying character to fall within the Telugu block (U+0C00–U+0C7F); lines with zero Telugu-block characters are discarded, while script-neutral characters (punctuation, numerals, whitespace) are preserved. This ensures that all downstream generation contexts contain authentic Telugu morphology rather than romanised or mixed-script artefacts. Segmentation proceeds via a sliding window of size $k = 5$ with stride $k = 5$, applied within individual movie boundaries to prevent narrative contamination across screenplay boundaries and to ensure that every evaluation instance is statistically independent. Each evaluation instance is a five-tuple $\langle c_1, c_2, c_3, c_4, r \rangle$ comprising a four-turn context and a gold reference response r. From the filtered corpus we draw 500 evaluation instances via interleaved stratified sampling (selecting every $\lfloor N/500 \rfloor$-th evaluation instance to preserve proportional movie representation), then apply a global shuffle at seed 42 to prevent ordering artefacts.

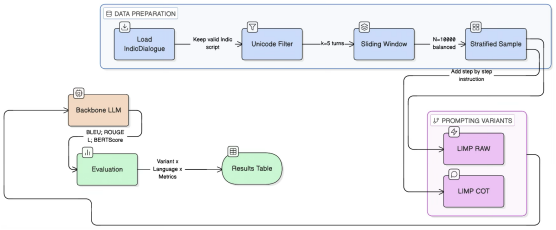


Figure 1: The end-to-end LIMP pipeline.

LIMP encodes linguistic and cultural expert knowledge as structured prompt constraints, replacing the model's implicit, and in low-resource Dravidian settings often absent, encoding of Telugu-

specific grammar with explicit inference-time instruction (Brown et al., 2020). Two strategies constitute the framework on a spectrum from dense simultaneous constraint injection to sequential analytical scaffolding.

3.0.1 LIMP-RAW: Dense Constraint Prompting

LIMP-RAW is a single, densely structured prompt of ≈480 tokens encoding five categories of linguistic and cultural constraints as hard generation rules:

C1. Honorific Register Rules. Verb-ending inventories: honorific (*-tunnaaru*, *-cheshaaru*) and familiar (*-tunnaavu*, *-cheshaavu*), keyed to speaker–addressee social rank (Steever, 1998).

C2. SOV Word Order. An explicit instruction that Telugu is a strict SOV language, with a contrastive correct/incorrect example illustrating the target syntax (Krishnamurti, 2003).

C3. Agglutinative Morphology. Post-positional suffix inventory for case markers (*-ki*, *-tho*, *-lo*, *-nundi*) with a rule prohibiting free-standing prepositions (Krishnamurti, 2003).

C4. Three-Dialect Discrimination. Diagnostic lexical markers distinguishing standard Coastal Andhra Telugu, colloquial Telangana/Hyderabadi Telugu (Urdu substrate), and Rayalaseema Telugu.

C5. Five Cinematic Genre Conventions. Genre-specific rhetorical shapes for action/confrontation, family drama, romance/*sringara*, comedy/*hasya*, and social/art-house dialogue.

All five constraint categories are delivered simultaneously; the model integrates them holistically during auto-regressive generation without an explicit reasoning trace. Output constraint: single spoken line, Telugu Unicode only (U+0C00–U+0C7F), no transliteration.

3.0.2 LIMP-COT: Six-Stage Analytical Scaffold

LIMP-COT extends the constraint vocabulary of LIMP-RAW into a sequential analytical chain-of-thought scaffold (Wei et al., 2022) of ≈920 tokens.

In Telugu dialogue, *rasa*, speaker hierarchy, and morphological register are interdependent: each must be committed to before the next can be resolved. LIMP-COT enforces this ordering through six analytically chained stages:

Stage 1: Rasa Identification. Identify dominant rasa from the nine Natyashastra categories (Bharata Muni, 1951) and characterise the emotional arc (escalating / de-escalating / pivoting). Rasa is the most upstream constraint (Sreejith et al., 2017).

Stage 2: Speaker Relationship and Power Axis. Map the speaker-addressee relationship onto hierarchical position (superior / equal / subordinate) and emotional alignment, then derive the mandatory Telugu honorific form (Steever, 1998).

Stage 3: Genre and Scene Type. Classify cinematic genre and name the scene type (e.g., villain-hero confrontation, mother-son reconciliation). Scene type determines rhetorical shape (Wierzbicka, 1991).

Stage 4: Linguistic Register. Commit to: (a) exact honorific verb endings; (b) target dialect (coastal Andhra / Telangana / Rayalaseema); (c) code-switching calibration. Explicit commitment prevents underspecification defaults (Jiang et al., 2020).

Stage 5: Narrative Function. Select exactly one of seven narrative functions (escalation, revelation, defiance, vulnerability, ironic pivot, *etc.*) to preserve rhetorical force (Chatman, 1980; Holtzman et al., 2020).

Stage 6: Final Generation. Synthesise all prior commitments into a single spoken Telugu line (Unicode-only, no transliteration) (Jiang et al., 2020).

The six stages are analytically interdependent: each takes the outputs of all prior stages as input, creating a commitment chain that distinguishes LIMP-COT from both generic CoT (Wei et al., 2022) and LIMP-RAW. The cost of this precision is reduced paraphrastic freedom, producing the semantic–lexical dissociation characterised in §5.

We evaluate all configurations with nine complementary metrics spanning the semantic–lexical–pragmatic measurement space. BERTSCORE F_1 (Zhang et al., 2020a) computes soft token-level matching via MURIL contextual embeddings (Khanuja et al., 2021), rewarding semantically equivalent paraphrases rather than penalising valid surface alternations. Cosine similarity operates on the same dense MURIL embeddings and measures high-level topical alignment in vector space. The character n-gram F-score (chrF) (**?**) provides character-level evaluation that is highly effective for agglutinative morphology, directly capturing partial suffix matches that word-level metrics miss entirely. BLEU (Papineni et al., 2002) and ROUGE-L are included as legacy baselines that reviewers expect, despite their well-documented failure modes for morphologically rich languages (Callison-Burch et al., 2006; Post, 2018; Mathur et al., 2020). Subword Jaccard similarity, $J(H, R) = |H \cap R|/|H \cup R|$, and the Dice coefficient, $D(H, R) = 2|H \cap R|/(|H| + |R|)$, measure lexical overlap over subword token sets and index morphological form fidelity (Bhat and Sharma, 2013; Sørensen, 1948). Distinct-1 and Distinct-2 (Li et al., 2016) measure the ratio of unique unigrams and bigrams in the generated output, penalising repetitive or degenerate generation.

All pairwise comparisons use paired t-tests and Wilcoxon signed-rank tests (Wilcoxon, 1945) at $N = 500$. Effect sizes are reported as Cohen's d (Cohen, 1988). Per-sample win rates provide a distribution-free summary statistic.

4 Experimental Setup

All experiments are implemented in Python 3.10 using `transformers` v4.40.0 (Wolf et al., 2020), `PyTorch` v2.3.0 (Paszke et al., 2019), and `scipy` v1.13.0 (Virtanen et al., 2020), running on a single NVIDIA RTX A6000 48 GB GPU and Weights & Biases experiment tracking (Weights & Biases, 2020). Models are loaded in FP16 with SDPA attention. **GEMMA-3-1B-IT** (Gemma Team, Google DeepMind, 2025) is a 1B-parameter instruction-tuned decoder-only transformer with a 128K token context window, evaluated under four configurations: Zero-Shot (raw context with no injected constraints), No-LIMP (raw context with a minimal continuation instruction but no linguistic framework), LIMP-RAW, and LIMP-COT. **SARVAM-1** (Sarvam AI, 2024) is a 2B Indic-focused decoder-only model with a 4,096-token effective context window, evaluated under three configurations: Zero-Shot, LIMP-RAW, and CoT-Compressed—a condensed ≈241-token version of the LIMP-COT scaffold designed to fit within SARVAM-1's context limit while preserving all six reasoning stages. All seven configurations decode greedily (`num_beams=1`, `do_sample=False`, `max_new_tokens=256`) to ensure reproducibility and to isolate prompt-structure effects from stochastic generation variance. MURIL (`google/muril-base-cased`) (Khanuja et al., 2021) serves as the BERTSCORE reference model for all semantic fidelity evaluations.

5 Results

We evaluate seven configurations on 500 Telugu evaluation instances. The primary finding is that LIMP-COT achieves 2× higher Jaccard and Dice than LIMP-RAW on the same backbone (Jaccard: 0.0140 vs. 0.0098; Dice: 0.0274 vs. 0.0193; $p < 0.001$), demonstrating that sequential analytical commitment to linguistic constraints produces more form-faithful Telugu than holistic constraint injection. Under LLM-as-judge evaluation, LIMP-COT achieves the highest Adequacy (3.90 ± 0.91) among GEMMA-3-1B-IT configurations, while LIMP-RAW achieves the highest Fluency (4.07 ± 0.93). The zero-shot baseline achieves the highest BERTSCORE F_1 (0.9760) across all configurations. This *semantic–lexical–pragmatic dissociation*, where no single configuration dominates across all metric classes, is a substantive finding: automatic semantic metrics, morphosyntactic surface fidelity, and pragmatic adequacy are orthogonal evaluation dimensions in agglutinative Telugu. Table 1 presents full quantitative results; Table 2 presents LLM-as-judge ratings.

LIMP-COT achieves Jaccard = 0.0140 and Dice = 0.0274, both higher than LIMP-RAW (Jaccard = 0.0098, Dice = 0.0193), with all differences statistically significant ($p < 0.001$, paired t-test and Wilcoxon (Wilcoxon, 1945)). This is consistent with Stage 4 of the CoT scaffold, which requires explicit commitment to specific verb endings and dialect markers before generation, anchoring the output to precise surface tokens in the gold reference. Absolute Jaccard and Dice values are low across all configurations, as expected: exact subword token matching in an agglutinative paradigm is combinatorially difficult even for semantically

Model	Strategy	Cosine↑	BS-F_1↑	chrF↑	BLEU↑	Jaccard↑	Dice↑	Dist-1↑	Dist-2↑
GEMMA-3-1B-IT	Zero-Shot	0.9919	**0.9760**	0.0782	**0.0184**	**0.0423**	**0.0760**	0.9514	0.7807
GEMMA-3-1B-IT	No-LIMP	**0.9921**	0.9712	**0.1006**	0.0080	0.0289	0.0547	0.8785	0.9787
GEMMA-3-1B-IT	LIMP-RAW	0.9889	0.9704	0.0069	0.0029	0.0098	0.0193	0.8963	0.9948
GEMMA-3-1B-IT	LIMP-COT	0.9899	0.9715	0.0205	0.0038	0.0140	0.0274	0.8923	0.9907
SARVAM-1	Zero-Shot	0.9919	0.9721	0.0706	0.0074	0.0247	0.0469	0.9309	0.9845
SARVAM-1	LIMP-RAW	0.9883	0.9709	0.0030	0.0037	0.0097	0.0191	0.9960	0.9997
SARVAM-1	CoT-Compressed	0.9885	0.9707	0.0017	0.0031	0.0077	0.0152	**0.9982**	**1.0000**

Table 1: Main results on 500 Telugu evaluation instances. **BS-F_1** = BERTScore F_1 computed with MURIL backbone. Bold = best per column. Note: BLEU, Jaccard, Dice, and chrF values are close to zero across all configurations, as expected for an agglutinative language where root words merge with post-positional suffixes, creating a combinatorially large surface paradigm that penalises exact n-gram and token overlap metrics.

Configuration	Adequacy	Fluency	Overall	κ
GEMMA-3-1B-IT Zero-Shot	3.80 ±0.98	3.80 ±1.05	**4.17 ±0.82**	0.130
GEMMA-3-1B-IT LIMP-RAW	3.60 ±0.84	**4.07 ±0.93**	3.83 ±0.82	0.079
GEMMA-3-1B-IT LIMP-COT	**3.90 ±0.91**	3.87 ±0.81	3.97 ±0.80	−0.003
SARVAM-1 LIMP-RAW	4.00 ±0.93	3.83 ±0.69	4.10 ±0.79	0.040
SARVAM-1 CoT-Comp	3.97 ±1.02	3.97 ±1.11	3.70 ±0.74	0.032

Table 2: LLM-as-judge evaluation on a stratified sample of 500 instances. Ratings are on a 1–5 integer scale (1 = unacceptable, 5 = excellent). κ = inter-judge Cohen's Kappa on the Overall dimension. LIMP-COT achieves the highest Adequacy among GEMMA-3-1B-IT configurations; LIMP-RAW achieves the highest Fluency.

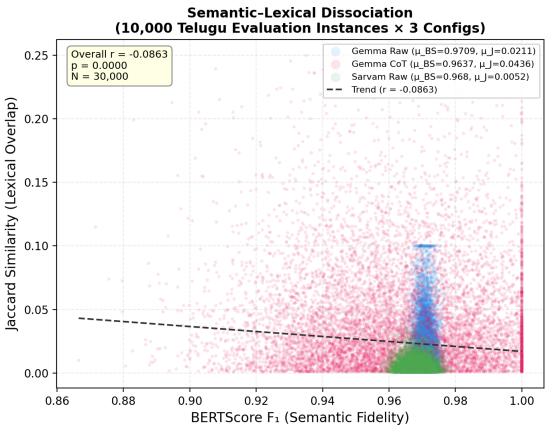


Figure 2: Visualisation of the semantic–lexical dissociation. Each point is one of the 500 evaluation samples. The absence of positive correlation between BERTSCORE F_1 and Jaccard confirms that the two metric classes capture orthogonal dimensions of Telugu dialogue generation quality.

correct responses (Bhat and Sharma, 2013; Mielke et al., 2021). The zero-shot baseline achieves the highest Jaccard (0.0423) and Dice (0.0760) overall, indicating that unconstrained generation more frequently produces surface-coincident outputs, while LIMP-prompted outputs diverge from reference surface forms in service of pragmatic correctness, a trade-off confirmed by the judge evaluation.

The LLM-as-judge results (Table 2) reveal a dissociation between fluency and adequacy. LIMP-RAW produces highly fluent Telugu (Fluency = 4.07, the highest among all configurations) but incurs a sharp drop in Adequacy (3.60), failing to anchor the output firmly to the conversational context. Injecting the CoT scaffold reverses this: LIMP-COT raises Adequacy to 3.90 while maintaining comparable Fluency (3.87), Cohen's Kappa values are in the slight-to-fair range across configurations ($\kappa = -0.003$ to 0.130), reflecting the inherent subjectivity of evaluating culturally-embedded cinematic Telugu; native speaker annotation remains the necessary validation step.

On the cross-model comparison, SARVAM-1 LIMP-RAW (2B, Indic-pretrained) achieves Jaccard = 0.0097 and Dice = 0.0191, comparable to GEMMA-3-1B-IT LIMP-RAW (Jaccard = 0.0098, Dice = 0.0193), while GEMMA-3-1B-IT LIMP-COT (Jaccard = 0.0140) outperforms both on lexical fidelity. This pattern is consistent with the hypothesis that structured linguistic scaffolding is a more productive lever than parametric scale for form-faithful generation, though the comparison remains partially confounded: SARVAM-1 CoT-Compressed uses a condensed scaffold rather than the full LIMP-COT prompt, and observed differences reflect the joint effect of model capability and prompt fidelity. LIMP-COT also exhibits substantially higher output variance on lexical metrics, attributable to reasoning-chain collapse when early stages err (e.g., misidentified rasa propagates through all downstream commitments) (Shi et al., 2023).

6 Conclusion

We introduced LIMP, a training-free prompting framework for Telugu multi-turn dialogue generation that injects rasa theory, honorific morphology, dialectal variation, and cinematic genre conventions at inference time, demonstrating that expert linguistic knowledge can be encoded in prompt structure without fine-tuning or additional compute.

The primary empirical finding is that LIMP-CoT achieves 2× higher Jaccard and Dice than LIMP-Raw on the same backbone (Jaccard = 0.0140 vs. 0.0098; $p < 0.001$), establishing that sequential analytical commitment to linguistic constraints produces more form-faithful Telugu than holistic constraint injection. Under LLM-as-judge evaluation, LIMP-CoT achieves the highest Adequacy (3.90 ± 0.91) while LIMP-Raw achieves the highest Fluency (4.07 ± 0.93), confirming that fluency and pragmatic adequacy are dissociable and that the CoT scaffold specifically addresses the adequacy deficit of dense constraint prompting. The zero-shot baseline achieves the highest automatic metric scores (BERTScore $F_1 = 0.9760$; chrF = 0.0782; Jaccard = 0.0423), establishing a strong floor and motivating the multi-dimensional evaluation framework: automatic metrics alone are insufficient to capture pragmatic quality in culturally-embedded Telugu dialogue. No single configuration dominates across all metric classes.

This work has five limitations that bound the scope of its findings. First, the absence of human evaluation by native Telugu speakers is a critical gap. All morphosyntactic and cultural claims, including rasa coherence and honorific correctness, are inferred from automatic metrics and proxy judge models and require human validation before strong interpretive claims can be made. Second, the cross-model comparison is partially confounded: Sarvam-1 is evaluated under a compressed CoT scaffold rather than the full LIMP-CoT prompt, so the observed lexical fidelity gap reflects the joint effect of model capability and prompt completeness and cannot be attributed to either factor in isolation. Third, rasa identification accuracy in Stage 1 is unverified. Errors may propagate through the full commitment chain and silently degrade output quality without detection by automatic metrics. Fourth, all results are domain-specific to subtitle-derived dialogue. The evaluation corpus is drawn from OpenSubtitles.org (Arnob et al., 2024) and may include dubbed foreign content alongside original Telugu productions. Dubbed material does not necessarily exhibit authentic *rasa* patterns or native honorific morphology, potentially reducing the cultural validity of both evaluation instances and gold references. This confound cannot be resolved without per-instance provenance metadata not currently available in the corpus, and generalisation to conversational, formal, or journalistic Telugu registers remains uncharacterised. Fifth, greedy decoding (`num_beams=1`) may systematically disadvantage LIMP-CoT, which benefits from diverse sampling (see §5). This experimental constraint should be ablated in future work. Priority future directions include human evaluation for honorific accuracy and rasa coherence, extension to Tamil, Kannada, and Malayalam (Lauscher et al., 2020), and evaluation at the 7B+ parameter scale.

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Hussain Muhammad, Kalika Bali, and Sunayana Sitaram. 2023. MEGA: Multilingual evaluation of generative AI. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267. Association for Computational Linguistics.

Noor Mairukh Khan Arnob, A. Faiyaz, Md Mubtasim Fuad, Shah Murtaza Rashid Al Masud, Baivab Das, and M.F. Mridha. 2024. IndicDialogue: A dataset of subtitles in 10 Indic languages for Indic language modeling. *Data in Brief*, 55:110690.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, and 1 others. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, volume 1, pages 675–718. Association for Computational Linguistics.

Bharata Muni. 1951. *The Nāṭyaśāstra: A Treatise on Hindu Dramaturgy and Histrionics*. Asiatic Society of Bengal, Calcutta. Translated by Manomohan Ghosh.

Riyaz Ahmad Bhat and Dipti Misra Sharma. 2013. Dependency parsing of Telugu: An agglutinative language. In *Proceedings of the 12th International Workshop on Treebanks and Linguistic Theories*, pages 197–207. Sofia University Press.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624. Association for Computational Linguistics.

Penelope Brown and Stephen C. Levinson. 1987. *Politeness: Some Universals in Language Usage*. Cambridge University Press, Cambridge.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Samuel Cahyawijaya, Holy Lovenia, Pascale Fung, and 1 others. 2023. NusaCrowd: Open source initiative for Indonesian NLP resources. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13745–13818. Association for Computational Linguistics.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of BLEU in machine translation research. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256. Association for Computational Linguistics.

Seymour Chatman. 1980. *Story and Discourse: Narrative Structure in Fiction and Film*. Cornell University Press, Ithaca, NY.

Jacob Cohen. 1988. *Statistical Power Analysis for the Behavioral Sciences*, 2nd edition. Lawrence Erlbaum Associates, Hillsdale, NJ.

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

David M. Eberhard, Gary F. Simons, and Charles D. Fennig. 2023. *Ethnologue: Languages of the World*, 26th edition. SIL International, Dallas, TX. Available online at https://www.ethnologue.com.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. Complexity-based prompting for multi-step reasoning. In *International Conference on Learning Representations*.

Jay Gala, Pranjal A. Kamble, Raj Dabre, Ratish Puduppully, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2023. IndicTrans2: Towards high-quality and accessible machine translation of all 22 scheduled Indian languages. *Transactions on Machine Learning Research*.

Gemma Team, Google DeepMind. 2025. Gemma 3 technical report. https://arxiv.org/abs/2503.19786. ArXiv:2503.19786.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what a language model knows? In *Transactions of the Association for Computational Linguistics*, volume 8, pages 423–438. MIT Press.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satya Golla, N Sai Chetan Gokul, Avik Bhatt, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pretrained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961. Association for Computational Linguistics.

Simran Khanuja, Divyanshu Kakwani, Satya Golla, Gokul Bhatt, Amir Arora, Anoop Kunchukuttan, and Partha Talukdar. 2021. MuRIL: Multilingual representations for indian languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7319–7330. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213.

Bhadriraju Krishnamurti. 2003. *The Dravidian Languages*. Cambridge University Press, Cambridge.

Anoop Kunchukuttan, Divyanshu Kakwani, Satya Golla, Avik Bhatt, Amir Arora, Mitesh M. Khapra, and Pratyush Kumar. 2020. The AI4Bharat-IndicNLP corpus: Monolingual corpora and word embeddings for Indic languages. In *Proceedings of the Eighth Workshop on Asian Translation*, pages 68–77. Association for Computational Linguistics.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot cross-lingual transfer with multilingual transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 4483–4499. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119. Association for Computational Linguistics.

Aman Madaan, Amrith Peng, Anshuman Roth, Dhruv Yang, Shrimai Prabhumoye, Paul Liang, Jamie Callan, and Alan W Black. 2020. Politeness transfer: A tag and generate approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1869–1881. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997. Association for Computational Linguistics.

Sabrina J. Mielke, Ryan Cotterell, Kyle Gorman, Brian Roark, and Jason Eisner. 2021. Between words and characters: A brief history of open-vocabulary modeling and tokenization in NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7557–7573. Association for Computational Linguistics.

Tong Niu and Mohit Bansal. 2018. Polite dialogue generation without parallel data. *Transactions of the Association for Computational Linguistics*, 6:373–389.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32.

Flor Miriam Plaza-del Arco, Roman Klinger, Rodrigo Agerri, and Valerio Basile. 2024. Emotion analysis in NLP: Trends, gaps and roadmap for future directions. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*. ELRA and ICCL.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191. Association for Computational Linguistics.

Vinodkumar Prabhakaran, Kalina Bhatt, Hannah Rashkin, and Mark Diaz. 2024. Power and politeness: Computational perspectives on social interaction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Tutorial Abstracts)*. Association for Computational Linguistics.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325. Association for Computational Linguistics.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 3118–3135. Association for Computational Linguistics.

Jivnesh Sandhan, Ashish Shukl, Bhargav Bhatt, and Pawan Goyal. 2023. Computational analysis of rasa in Sanskrit poetry. In *Proceedings of the 2023 Conference of the European Chapter of the Association for Computational Linguistics (Student Research Workshop)*, pages 81–90. Association for Computational Linguistics.

Sarvam AI. 2024. Sarvam-1: A comprehensive language model for Indian languages. https://www.sarvam.ai/blog/sarvam-1. Accessed: March 2025.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative models: Experiments on the Ubuntu dialogue corpus. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, pages 31210–31227. PMLR.

Thorvald Sørensen. 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content. *Kongelige Danske Videnskabernes Selskab*, 5(4):1–34.

V. Sreejith, C. K. Aravindh, and Priyanka Prabhu. 2017. Navarasa: Understanding emotions in poetry. In *Proceedings of the 14th International Conference on Natural Language Processing*, pages 261–271. NLP Association of India.

Sanford B. Steever, editor. 1998. *The Dravidian Languages*. Routledge, London and New York.

Karumuri V. Subbarao. 2012. *South Asian Languages: A Syntactic Typology*. Cambridge University Press, Cambridge.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, and 1 others. 2020. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Weights & Biases. 2020. Weights & biases: Machine learning experiment tracking. https://wandb.ai. Accessed: 2025.

Anna Wierzbicka. 1991. *Cross-Cultural Pragmatics: The Semantics of Human Interaction*. Mouton de Gruyter, Berlin.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and 1 others. 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. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. BETO, Bentz, Becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 833–844. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020a. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020b. DIALOGPT: Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278. Association for Computational Linguistics.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *International Conference on Learning Representations*.

TamilMayangoliSpell: An Open-Source Neural Framework for Context-Sensitive Mayangoli Error Correction in Tamil

Yazhmozhi VM[1] **Annalu Waller**[2] **Jacky Visser**[3]
[1]Computing, Edinburgh Napier University International College
[2]School of Science, Engineering and Environment, University of Salford
[3]Computing, University of Dundee
Correspondence: y.murugesan@napier.ac.uk

Abstract

Mayangoli errors are context-sensitive errors in Tamil that arise from confusion among phonetically similar graphemes (e.g., ல/ள/ழ, ர/ற, ந/ன/ண). These errors are challenging for conventional spell checkers because both incorrect and correct forms are valid dictionary words, making dictionary lookup insufficient and requiring contextual modelling. We present **TamilMayangoliSpell**, a reproducible framework for Mayangoli error correction that combines (i) Tamil-specific preprocessing for sentence segmentation and normalisation, (ii) linguistically grounded error induction for generating training data constrained by dictionary validity, and (iii) fine-tuning of multilingual sequence-to-sequence models. Using 30,000 sentence pairs derived from TamilCorp, a massive multi-genre Tamil corpus and split 80/10/10 into train/validation/test, we fine-tune mBART, mT5, and NLLB under a small hyperparameter grid using greedy decoding with a maximum sequence length of 128. mT5 achieves the best performance (BLEU 99.28; Exact Match Accuracy 93.50%) and remains strong in a cross-genre evaluation on short stories. The preprocessing scripts, generated parallel datasets, and trained models are publicly available in a GitHub repository.

1 Introduction

Context-sensitive real-word errors are valid dictionary words that become incorrect in a given context. Such errors are non-trivial to identify in morphologically rich and inflectional languages like Tamil without context modelling. This is concerning for *Mayangoli* errors: context-sensitive errors caused by confusion among phonetically similar graphemes (e.g., ல/ள/ழ, ர/ற, ந/ன/ண). A dictionary might contain குளம் (pond) as well as குலம் (clan). When a Tamil writer has used one for the other in an unrelated context, this is an instance of a Mayangoli error.

Existing works on the detection and correction of Mayangoli errors in Tamil have predominantly used dictionary-based or statistical methods (Segar and KengatharaiyerSarveswaran, 2015; Sithamparanathan and Uthayasanker, 2019), with limited reproducibility due to the lack of publicly available datasets and implementations (Sampath and Shanmugavel, 2023; Rajalakshmi et al., 2023). To address this gap, we propose **TamilMayangoliSpell**, an open-source neural framework for Mayangoli error correction that combines a linguistically grounded error-induction pipeline with multilingual sequence-to-sequence (Seq2Seq) models.

Contributions:

- A linguistically grounded error induction pipeline for generating Tamil context-sensitive Mayangoli errors using confusion-group substitutions constrained by dictionary validity.
- A controlled empirical comparison of multilingual Seq2Seq models (mBART, mT5, and NLLB) for Tamil orthographic error correction under identical training and decoding settings.
- Cross-genre generalisation, demonstrating that the learnt correction behaviour transfers beyond the training domain.
- Code, preprocessing scripts, trained models, and experiment configurations are published on GitHub[1] to support reproducible research in Tamil NLP.

2 Background

Classical Tamil grammar books like the *Tholkaappiyam* (Pillai, 1975) describe phonological distinctions, explaining confusions between graphemes.

[1]https://github.com/TamilGeekGirl/TamilMayangoliSpell

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 42–51
July 3, 2026 ©2026 Association for Computational Linguistics

The major types of context sensitive errors in Tamil are summarised in this section, with a focus on Mayangoli errors, which form the scope of our work.

2.1 Context-sensitive errors

Context-sensitive errors are valid words that are irrelevant to the context in which they are used. They differ from non-word errors (words not in the lexicon/dictionary), which are often detectable via dictionary lookup. In Tamil, context-sensitive errors can go undetected because Tamil orthography supports many pairs that are both valid dictionary words and differ by a single character.

Context-sensitive errors in Tamil include *Kuril–Nedil* errors (short vs. long vowel confusions) and *Mayangoli* errors.

2.2 Kuril–Nedil errors

Kuril–Nedil errors involve replacing a short vowel character with its long counterpart or vice versa, sometimes yielding another valid word in the dictionary. For example, the word படம் (Transliteration: padam, Translation: picture) might be used instead of the word பாடம் (Transliteration: paadam, Translation: lesson). They are known as Kuril Nedil errors. Both படம் and பாடம் are valid words in the dictionary and might be used in an irrelevant context.

2.3 Mayangoli errors

Mayangoli errors arise from confusions among consonant graphemes associated with similar phonemes. The confusion groups in Tamil are as follows:

1. **'la' group:** ல/ள/ழ (Transliteration: la)
2. **'ra' group:** ர/ற (Transliteration: ra)
3. **'na' group:** ந/ன/ண (Transliteration: na)

These confusions also appear in UyirMei (consonant+vowel) combinations. A key practical challenge is that a Tamil writer might get confused and use them in irrelevant contexts. This work focuses exclusively on Mayangoli errors.

3 Related Work

In Tamil, spelling correction has been studied through dictionary-based, rule-based, statistical, and neural approaches. Dictionary-based methods remain effective for non-word errors but are weak for context-sensitive real-word errors where both candidates are valid words (Uthayamoorthy et al., 2019). Statistical approaches have been proposed for Tamil spelling correction (Sakuntharaj and Mahesan, 2016, 2017, 2018), typically modelling error likelihood and contextual probabilities; however, they often struggle with long-range dependencies and semantic appropriateness.

Neural methods have achieved strong results for context-aware correction, including transformer-based approaches and text-to-text models (Rajalakshmi et al., 2023; Elango and Pati, 2023). Yet, low-resource conditions for Tamil (limited task-specific annotated corpora) constrain model training and evaluation, and the lack of publicly available resources limits reproducibility. Multilingual pretrained encoder–decoder models such as mBART (Liu et al., 2020), mT5 (Xue et al., 2021), and NLLB (NLLBTeam et al., 2024) offer cross-lingual transfer that can benefit low-resource languages, and are well-suited for Seq2Seq transformations like spelling correction and text normalisation. Our work builds on these advances by providing a reproducible Mayangoli-specific pipeline and an empirical comparison of multilingual Seq2Seq models under controlled induction and cross-genre evaluation.

4 Data and Methodology

An overall pipeline of the proposed TamilMayangoliSpell framework is illustrated in Figure 1. The process starts with the raw Tamil corpora and is followed by abbreviation normalisation and text cleaning to avoid irregular sentence segmentation. Then, Mayangoli-specific substitution rules are used to produce linguistically plausible error patterns, constrained by dictionary membership. These are used to construct parallel sentence pairs for supervised training. Finally, multilingual Seq2Seq models are fine-tuned on the generated dataset and evaluated using both intrinsic metrics and cross-genre testing.

Algorithms 1–4 are used in the core pipeline for the creation of the parallel dataset. These algorithms are provided for clarity and reproducibility. The actual implementation of all the algorithms is available in the GitHub repository.[2]

[2] https://github.com/TamilGeekGirl/TamilMayangoliSpell/tree/main/Preprocessing%20scripts

4.1 Corpus source

TamilCorp (Yazhmozhi VM and Waller, 2025) is a balanced Tamil corpus comprising approximately 1.69 billion tokens collected from 17 genres, including newspapers, literature, government documents, and web content. The corpus is designed to ensure broad genre coverage—letters, newspapers, law, proverbs, essays, reviews, arts, magazines, finance, medicine, research journals, government, short stories, textbooks, other books, religious texts, and Wikipedia—drawing inspiration from established balanced corpora like Brown (Brown University, 2024) and the British National Corpus (British National Corpus, 2024). Data was obtained through web scraping and publicly available documents, with additional permissions secured from selected publishers, and collected in compliance with UK copyright regulations. A small chunk of the newspaper genre of TamilCorp (Yazhmozhi VM and Waller, 2025) was used as a text source for preprocessing and error induction. The resulting error induced parallel dataset is intended to represent context-sensitive errors where both erroneous and corrected forms are valid words.

4.2 Preprocessing

Preprocessing ensures that raw Tamil text is cleaned, normalised, and segmented into linguistically valid sentences. A major practical issue is sentence boundary detection. Abbreviations (e.g., month abbreviations ending with period, initials in names) can trigger false sentence splits. To remedy this, common abbreviations are normalised by expanding them prior to sentence splitting and cleaning. Algorithms 1 and 2 summarise the preprocessing pipeline: Algorithm 1 normalises Tamil text by expanding abbreviations and preventing incorrect sentence boundary detection, whereas Algorithm 2 implements memory-efficient chunk-based ingestion and preprocessing of large Tamil corpora into sentence-level units.

4.3 Error induction overview

Supervised training pairs are created by injecting Mayangoli errors into clean sentences:

$$\langle x_{\text{err}}, x_{\text{clean}} \rangle$$

For each word containing a target character (including UyirMei), candidate substitutions are generated within the same confusion group and only those where the substituted word exists in a dictionary (contains 2,583,001 words) are retained.[3] This ensures induced errors are context-sensitive real-word errors rather than non-words. Algorithms 3 and 4 summarise the Mayangoli error induction pipeline: Algorithm 3 constructs linguistically valid Mayangoli substitution mappings for all UyirMei characters and Algorithm 4 generates supervised context-sensitive Mayangoli error–correction sentence pairs using dictionary-constrained substitutions.

Algorithm 1 Abbreviation Normalisation

Require: Tamil text segment T, abbreviation map A
Ensure: Normalised text T'
1: $T' \leftarrow T$
2: **for all** $(abbr, full) \in A$ **do**
3: Replace occurrences of $abbr$ in T' with $full$
4: **end for**
5: Preserve multi-dot abbreviations to avoid sentence boundary errors
6: **return** T'

4.4 Advantages of the pipeline

The proposed pipeline has a few advantages for processing low-resource languages. They are as follows:

- It supports scalable dataset generation without manual annotation.
- Confusion groups are linguistically grounded so that the induced errors are linguistically plausible and context-sensitive.
- The modularity of our approach facilitates adopting this method to other types of errors in Tamil or another Dravidian language.
- The pipeline is combined with multilingual pretrained models to facilitate effective training with less supervised data.

4.5 Dataset size and split

The full pipeline yields 279,837 candidate pairs. We subsample 30K pairs to balance computational cost and training efficiency. We split this subset into 80/10/10 train/validation/test at the sentence-pair level to avoid overlap across splits.

[3] https://github.com/KaniyamFoundation/all_tamil_words/tree/master

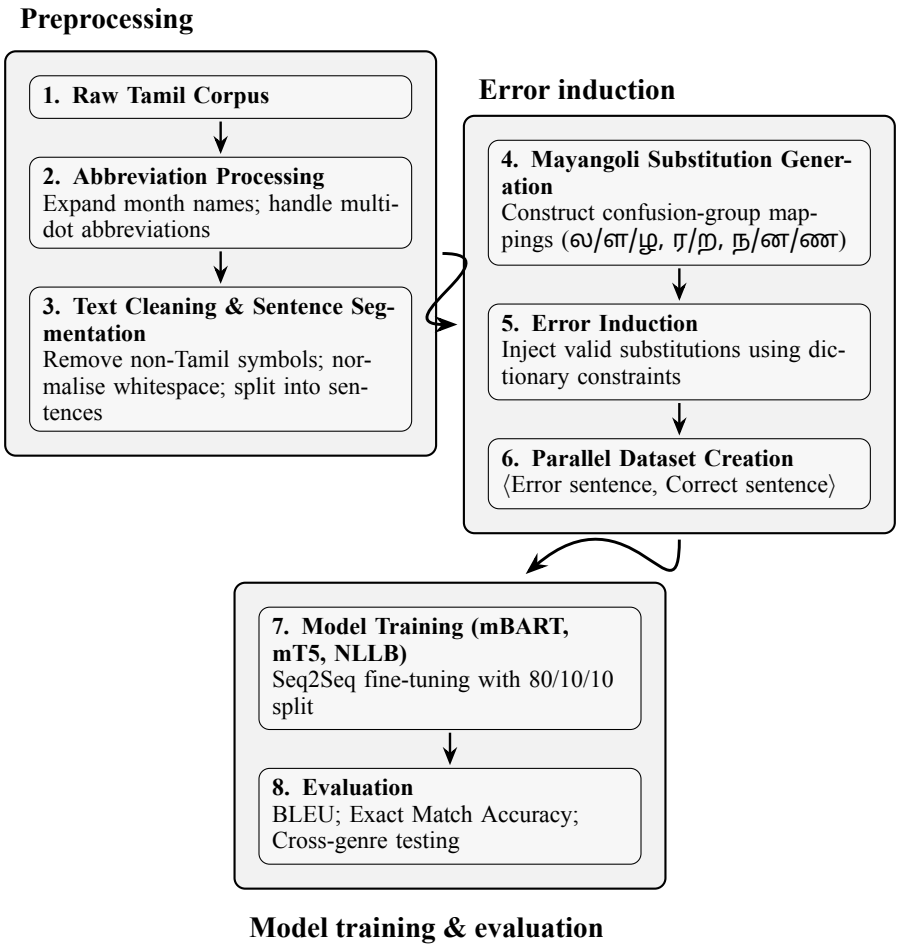


Figure 1: Overview of the TamilMayangoliSpell pipeline.

Algorithm 2 Sentence Extraction and Cleaning

Require: Raw Tamil corpus C, chunk size k
Ensure: List of cleaned sentences S

Initialise empty list S
while more text in C **do**
  Read next chunk c of size k
  $c \leftarrow \text{AbbreviationNormalisation}(c)$
  Remove non-Tamil symbols and normalise whitespace
  Segment c into sentences using Tamil-aware rules
  Append sentences to S
end while
return S

Algorithm 3 Mayangoli Substitution Generation

Require: Set of Tamil UyirMei characters U
Ensure: Substitution set R

Initialise empty set R
for all $u \in U$ **do**
  Decompose u into vowel and consonant components (v, c)
  if c belongs to a confusion group **then**
    for all $c' \neq c$ in same group **do**
      Generate $u' \leftarrow (v, c')$
      Add (u, u') to R
    end for
  end if
end for
return R

5 Models

Multilingual sequence-to-sequence (Seq2Seq) models are a natural choice for context-sensitive spelling correction, since they can be trained to map an incorrect sentence directly to its corrected form (Lewis et al., 2020; Liu et al., 2020). In this work, we focus on models that satisfy several practical requirements. Firstly, the model should include Tamil in its pretraining data so that its representations are compatible with the script (Conneau et al., 2020; Xue et al., 2021;

Algorithm 4 Error Induction and Dataset Construction

Require: Clean sentences S, dictionary D, substitutions R
Ensure: Dataset P of ⟨error, correct⟩ pairs

```
Initialise empty list P
for all s ∈ S do
    for all word w in s do
        for all (x, y) ∈ R do
            if x occurs in w then
                w′ ← replace x with y in w
                if w′ ∈ D then
                    Create s′ by replacing w with w′ in s
                    Add (s′, s) to P
                end if
            end if
        end for
    end for
end for
return P
```

NLLBTeam et al., 2024). Secondly, we restrict our choice to encoder–decoder architectures, as they are better suited for generating corrected text compared to models designed for token-level prediction (Lewis et al., 2020; Liu et al., 2020). Thirdly, we consider models that use subword tokenisation methods such as SentencePiece or Byte Pair Encoding (BPE), which are known to handle morphologically rich languages effectively (Kudo and Richardson, 2018; Sennrich et al., 2016). Finally, we prioritise models that have already shown strong performance in multilingual generation tasks, especially in low-resource settings (Liu et al., 2020; Xue et al., 2021; NLLBTeam et al., 2024).

In light of the above, we selected mBART (Liu et al., 2020), mT5 (Xue et al., 2021) and NLLB (NLLBTeam et al., 2024) for our experiments. The differences in pretraining objectives, vocabulary formation and handling of multilingual data makes them ideal for a comparative analysis with respect to Mayangoli error correction. We did not include models that use only the encoder like mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020), as they are intended for classification or mask prediction, so not well suited to sequence-to-sequence correction.

6 Experimental Setup

6.1 Training details

We fine-tuned all the models for 5 epochs using mixed precision (FP16) with early stopping (patience=2, based on validation loss) and saved the best checkpoint per run, ensuring convergence without overfitting. Validation loss stabilised within this range across models, consistent with prior findings on the efficient fine-tuning of pretrained Seq2Seq models (Popel and Bojar, 2018; Dodge et al., 2020; Mosbach et al., 2020). Hyperparameters are selected via a small grid:

- Batch size: $\{4, 8\}$
- Learning rate: $\{3 \times 10^{-5}, 1 \times 10^{-4}\}$

These parameters were selected because they are consistently reported as the chief factors affecting neural sequence-to-sequence optimisation dynamics and model performance (Smith, 2018; Popel and Bojar, 2018; Mosbach et al., 2020). Other hyperparameters (e.g., dropout, weight decay, and warm-up steps) were set to their default values as provided by the underlying framework, since relevant literature indicates that they have relatively smaller effects on convergence and generalisation (Devlin et al., 2019; Liu et al., 2020).

6.2 Decoding

During inference, greedy decoding is used with a maximum sequence length of 128 for all models to ensure consistent decoding conditions.

6.3 Metrics

We report:

- **BLEU** (Papineni et al., 2002), which measures n-gram overlap.
- **Exact Match Accuracy (EMA)** (Rajpurkar et al., 2016), a strict metric requiring full-string equality.

EMA is most suited for spelling correction tasks, as it is stricter; even an error in a single character invalidates the correction. However, BLEU is also selected as an evaluation metric to capture partial correctness and to gain insight into the behaviour of the model when there are no exact matches.

7 Results

7.1 Main results on newspaper data

Table 1 summarises validation performance for each model under the hyperparameter grid. Overall, mT5 performs the best, achieving BLEU 99.28 and EMA 93.50%. mBART is stable across hyperparameter choices but lags in peak accuracy. NLLB achieves competitive BLEU but yields an EMA of 0 across all configurations.

mBART exhibits identical performance across hyperparameter configurations, suggesting stable convergence under the limited hyperparameter grid and dataset size (30K). This indicates low sensitivity to optimisation settings in this regime, consistent with observations in low-resource fine-tuning of pretrained Seq2Seq models (Dodge et al., 2020; Mosbach et al., 2020).

7.2 Cross-genre evaluation

As mT5 achieved the best overall performance (in newspaper genre), we tested its generalisation ability by evaluating it on a different genre (short stories; 30K) using the same hyperparameter grid. Results in Table 2 show consistently high BLEU ($\geq$ 98.87) and EMA ($\geq$ 89.60), with peak BLEU 99.28 and peak EMA 93.50.

Cross-genre evaluation yields identical scores to in-domain validation across all configurations. We verified that the datasets are disjoint and that no data leakage occurs. This behaviour may be attributed to the controlled error induction process, which produces similar distributions of Mayangoli confusions across genres, resulting in consistent task difficulty.

The cross-genre evaluation suggests that the learned correction behaviour generalises beyond the training domain, although real-world deployment may involve noisier and more diverse error patterns.

8 Analysis and Discussion

While the underlying models are standard, our contribution lies in a linguistically grounded error induction framework for Tamil orthography. Unlike generic noise injection, we enforce phonologically valid, dictionary-constrained substitutions to generate context-sensitive real-word errors. We intentionally use standard pretrained Seq2Seq architectures to isolate the effects of data generation and task formulation rather than architectural variation.

Although evaluation relies on synthetic induction, substitutions are constrained by phonological confusion groups and dictionary validity, producing linguistically plausible context-sensitive real-word errors. Unlike dictionary-based or surface-form approaches, Seq2Seq models capture sentence-level contextual dependencies required for resolving context-sensitive real-word Mayangoli confusions.

8.1 Why does mT5 perform best?

We attribute mT5's advantage to (i) its uniform text-to-text objective that naturally matches correction as "translate erroneous $\rightarrow$ correct", and (ii) SentencePiece tokenisation that often better handles subword composition in morphologically rich scripts. Empirically, mT5 exhibits both high BLEU and high EMA, indicating that improvements are not merely n-gram overlap but full-sequence correctness.

8.2 NLLB: high BLEU, zero EMA

NLLB yields a non-trivial BLEU yet a 0 EMA, indicating that the outputs share partial n-gram overlap but fail to achieve exact correction. We observed this pattern consistently across hyperparameter configurations and decoding settings (greedy decoding, max_length=128). A plausible explanation is that the model produces near-miss variants (e.g., correct context but wrong character choice or minor token boundary artifacts) that preserve many n-grams but fail strict equality. This highlights that BLEU can be misleading for orthographic correction and motivates stricter and character-level metrics.

8.3 Additional evaluation considerations

EMA is strict and may under-credit partial improvements (e.g., correcting one of multiple errors). Conversely, BLEU may over-credit near-misses. In future work, character-level F-score, edit distance, minimal-edit accuracy, and human evaluation could provide more nuanced conclusions for end-user applications.

9 Deployment and Practical Usage

The trained Mayangoli error correction models can be used as an end-user application to help Tamil writers and people with language difficulties, such as dyslexia. The modular architecture of the system enables future work to integrate it easily into web-based or mobile applications. We provide two

Table 1: Validation performance of multilingual Seq2Seq models on the newspaper dataset (30K sentence pairs). Best configuration per model is highlighted in bold.

Model	Batch	Epochs	LR	BLEU ↑	Exact Match (%) ↑
mBART	4	5	3×10^{-5}	**92.98**	**89.53**
	4	5	1×10^{-4}	**92.98**	**89.53**
	8	5	3×10^{-5}	**92.98**	**89.53**
	8	5	1×10^{-4}	**92.98**	**89.53**
mT5	4	5	3×10^{-5}	98.96	90.52
	8	5	3×10^{-5}	98.87	89.60
	4	5	1×10^{-4}	99.27	**93.50**
	8	5	1×10^{-4}	**99.28**	93.38
NLLB	4	5	3×10^{-5}	93.04	0.00
	8	5	3×10^{-5}	**94.73**	0.00
	4	5	1×10^{-4}	90.77	0.00
	8	5	1×10^{-4}	93.63	0.00

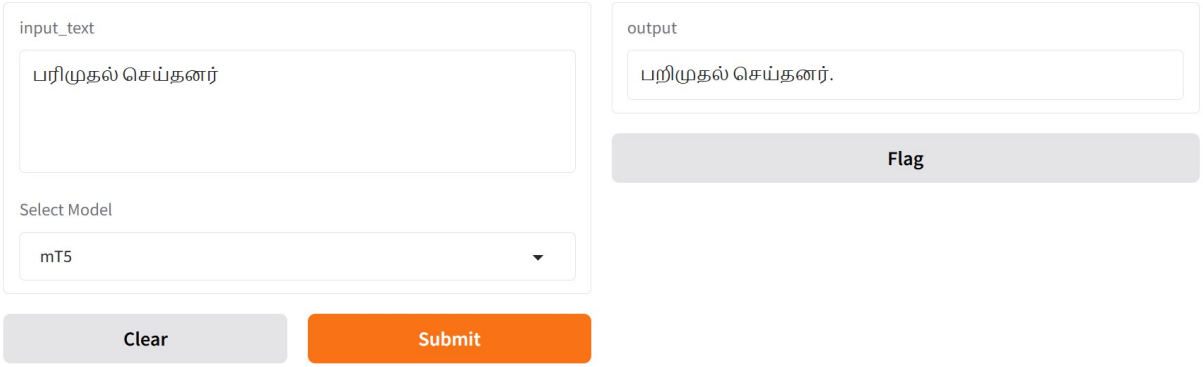


Figure 2: GUI-based Mayangoli error correction (mT5).

Table 2: Cross-genre evaluation of mT5 on Tamil short stories. Best scores are highlighted in bold.

BS	Ep	LR	BLEU ↑	EMA ↑
4	5	3×10^{-5}	98.96	90.52
8	5	3×10^{-5}	98.87	89.60
4	5	1×10^{-4}	99.27	**93.50**
8	5	1×10^{-4}	**99.28**	93.38

lightweight local deployment interfaces. The CLI is beneficial for developers and for integration into larger NLP pipelines. The GUI is beneficial for non-technical users. It is also well-suited for educational tools and writing aids.

Figures 2 and 3 illustrate the GUI-based deployment of Mayangoli error correction using the best-performing mT5 and mBART models. Figure 2 demonstrates the correction of a shorter sentence with two words, whereas Figure 3 demonstrates the correction of a longer sentence with five words. Additional deployment screenshots are available in the GitHub repository.

10 Limitations

Scope limitation. We focus exclusively on *Mayangoli* confusions and do not address other Tamil spelling error classes such as Kuril–Nedil errors, sandhi-related orthographic changes, non-word errors, or code-mixed writing. The system should therefore not be interpreted as a complete Tamil spell-checking solution.

Synthetic data limitation. Training and evaluation rely on synthetic error induction. Although we constrain substitutions by dictionary validity to ensure context-sensitive real-word errors, the induced distribution may not be fully representative of naturally occurring user errors (e.g., frequency of confusions, multi-error sentences, informal spelling, dialectal variation). As a result, reported scores may be optimistic relative to real-world text.

Compute and search limitation. Due to com-

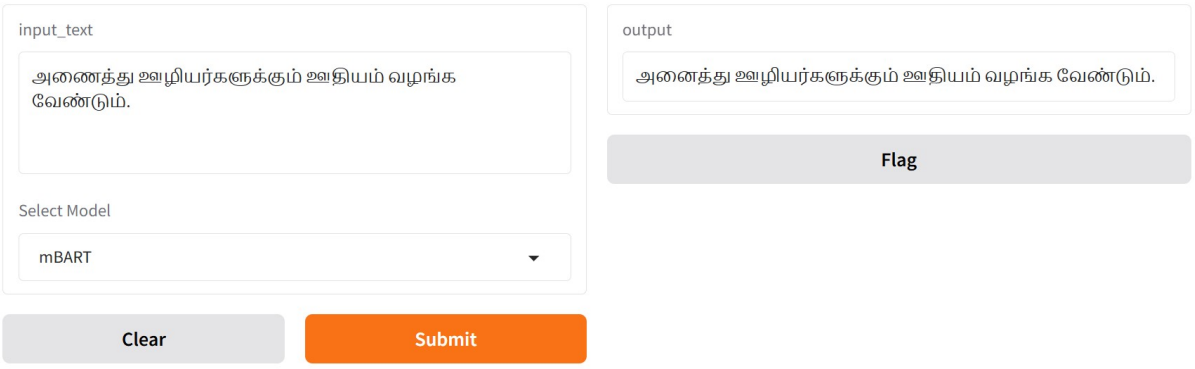


Figure 3: GUI-based Mayangoli error correction (mBART).

putational constraints, we train on a subsample of 30,000 sentence pairs and explore only a limited hyperparameter grid. Larger-scale fine-tuning using the full **TamilCorp** dataset (Yazhmozhi VM and Waller, 2025), additional model sizes, or broader optimisation strategies may change absolute performance and potentially alter model rankings.

Metric limitation. We evaluate using BLEU and EMA, which do not fully capture user-centric utility (partial corrections, edit distance reduction, or acceptability when multiple corrections are plausible). Metrics such as character error rate (CER) and edit distance are more suitable for orthographic correction and will be incorporated in future work.

Decoding limitation. We use a fixed decoding configuration (greedy decoding, max_length=128) for consistency. Alternative decoding settings may affect exact-match performance, especially for models sensitive to generation dynamics.

11 Conclusion

This paper explores how multilingual sequence-to-sequence (Seq2Seq) models can be used to correct *Mayangoli errors*, which are a category of context-sensitive errors in Tamil. We fine-tune three models—mBART, mT5, and NLLB—on a dataset of 30,000 sentence pairs derived from the newspaper genre of TamilCorp, a balanced Tamil corpus. To support this, we design a preprocessing and error-induction pipeline that generates linguistically plausible Mayangoli error-correction pairs.

Across experiments, all three models achieve strong BLEU scores under different hyperparameter settings. Among them, mT5 performs the best, achieving a BLEU score of 99.28 and an Exact Match Accuracy (EMA) of 93.50% (a batch size of 8, learning rate of 1×10^{-4}, and 5 training epochs). Performance of mBART over various hyperparameter settings is stable (BLEU: 92.98, EMA: 89.53%). This indicates that mBART may be more robust to hyperparameter tuning. However, it does not reach the same peak performance as mT5. NLLB generates strong BLEU scores but does not achieve exact match accuracy. This suggests high n-gram overlap does not guarantee correct full-sequence prediction.

Overall, the findings demonstrate that mT5 performs strongly for Mayangoli error correction in Tamil. This may result from its text-to-text formulation and subword tokenization capturing Tamil's morphological variation effectively. More generally, these results reveal that multilingual Seq2Seq models can be powerful tools for correcting context-sensitive spelling errors in Tamil. Future work involves generalising this approach for other error types in Tamil and evaluating performance in real-world usage settings. With more domain adaptation and user-based evaluation, **TamilMayangoliSpell** could substantially improve the accuracy and usability of Tamil language technology.

References

British National Corpus. 2024. Bnc homepage. http://www.natcorp.ox.ac.uk/. Accessed: 2024-10-20.

Brown University. 2024. Brown corpus manual. http://korpus.uib.no/icame/manuals/BROWN/INDEX.HTM. Accessed: 2024-10-20.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco

Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8440–8451.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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. Association for Computational Linguistics.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Vaan Amuthu Elango and Peeta Basa Pati. 2023. Tamil text error correction with multi-lingual t5 model. In *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)*, pages 1–6.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. 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. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2020. On the stability of fine-tuning BERT: misconceptions, explanations, and strong baselines. *CoRR*, abs/2006.04884.

NLLBTeam, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2024. Scaling neural machine translation to 200 languages. *Nature*, 630(26):841–846.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

K.R. Pillai. 1975. *Tolkappiyam: Text and Commentary*. University of Madras Press, Chennai, India.

Martin Popel and Ondřej Bojar. 2018. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

Ratnavel Rajalakshmi, Varsha Sharma, and Anand Kumar M. 2023. Context sensitive tamil language spellchecker using roberta. In *Speech and Language Technologies for Low-Resource Languages*, pages 51–61, Cham. Springer International Publishing.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Ratnasingam Sakuntharaj and Sinnathamby Mahesan. 2016. A novel hybrid approach to detect and correct spelling in tamil text. In *2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*, pages 1–6. IEEE.

Ratnasingam Sakuntharaj and Sinnathamby Mahesan. 2017. Use of a novel hash-table for speeding-up suggestions for misspelt tamil words. In *2017 IEEE International Conference on Industrial and Information Systems (ICIIS)*, pages 1–5. IEEE.

Ratnasingam Sakuntharaj and Sinnathamby Mahesan. 2018. Detecting and correcting real-word errors in tamil sentences. *Ruhuna Journal of Science*, 9(2):150–159.

Anbukkarasi Sampath and Varadhaganapathy Shanmugavel. 2023. Hybrid tamil spell checker with combined character splitting. *Concurrency and Computation: Practice and Experience*, 35(1):e7440.

Jananie Segar and KengatharaiyerSarveswaran. 2015. Contextual spell checking for tamil language. In *14th Tamil Internet 2015 Conference, Singapore.*, pages 379–383.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Lakshikka Sithamparanathan and Thayasivam Uthayasanker. 2019. A sinhala and tamil extension to generic environment for context-aware correction. In *2019 National Information Technology Conference (NITC)*, pages 102–106. IEEE.

Leslie N. Smith. 2018. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*.

Keerthana Uthayamoorthy, Kirshika Kanthasamy, Thavarasa Senthaalan, Kengatharaiyer Sarveswaran, and Gihan Dias. 2019. Ddspell - a data driven spell checker and suggestion generator for the tamil language. In *2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer)*, pages 1–6. IEEE.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.

Yazhmozhi VM and Annalu Waller. 2025. Building a balanced tamil corpus: Eda and lexical diversity comparison with english. In *2025 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)*, pages 1–6.

TamilTok: Morphologically-Informed Tokenization for Tamil

Surendhar Muthukumar and Aaricia Herygers and Lisa Beinborn
Institute of Computer Science, University of Goettingen, Germany
surendhar.m@stud.uni-goettingen.de, firstname.lastname@uni-goettingen.de

Abstract

Tokenization is fundamental to neural language modeling, yet for Tamil it remains largely adapted from general-purpose multilingual models without systematic consideration of the rich agglutinative morphology. We introduce TamilMorph, a large-scale dataset of more than 480,000 morphologically segmented Tamil word forms. Building on this new resource, we develop TamilTok, a morphology-aware tokenization framework that incorporates explicit morpheme structure into tokenizer training. We benchmark Tamil tokenization quality across multiple tokenization algorithms and vocabulary configurations and find that our approach improves both morphological alignment and downstream performance compared to previous approaches. Our morphological resource for Tamil and our systematic empirical analyses can guide future developments of tokenization for morphologically rich languages.

1 Introduction

Tokenization shapes the interface between raw symbolic text and computational representations in large language models. The choices of the tokenization algorithm and the granularity of the vocabulary determine how textual input is segmented into machine-learnable units, affecting model efficiency, preservation of semantic information, and generalization capabilities (Petrov et al., 2023). Subword tokenization has been optimized for the rather shallow morphological structure of English, and it remains an open research question how well the frequency-driven segmentation approach captures more complex agglutinative processes (Ahia et al., 2023; Arnett and Bergen, 2025).

Tamil is a Dravidian language spoken by more than 80 million people in India, Sri Lanka, Singapore, Malaysia, and communities of the global diaspora. It is written in an abugida script that encodes morphophonemic processes known as sandhi phenomena: systematic sound changes when two morphemes are combined. Tamil morphology is agglutinative, which means that suffixes can be stacked and the combined meaning can be derived systematically from the meaning of parts due to direct form–function correspondences. For example, the word விருந்தினர்களுக்கும் ("for the guests as well") is an agglutinative form derived from the root விருந்தினர் ("guest"). It is formed by sequentially adding the plural marker -கள் and the dative marker -உக்கும் (which additionally encodes the meaning "also/too").

Tamil uses systematic suffixation to inflect nouns across multiple grammatical cases and encodes tense, aspect, and agreement of verbs via sequential morpheme addition. Due to the high morphological regularity, a single lexical root can generate dozens of valid surface forms. The productive stacking of suffixes presents a challenge for frequency-driven subword tokenization, which often fragments morphemes into frequent surface forms, leading to disrupted morphological boundaries, especially in multilingual models (Bayram et al., 2025; Chintha and Konduru, 2025; Jabbar, 2023; Parra, 2024).

Despite Tamil's linguistic richness and large speaker base, dedicated studies on Tamil tokenization are scarce (Krishnan and Ragavan, 2021; T K et al., 2024). In practice, Tamil is usually processed with multilingual tokenizers, rather than dedicated monolingual ones. The design assumptions of multilingual models are largely shaped by high-resource analytic languages such as English, which are structured by whitespaces and exhibit only limited inflection. As a consequence, a disproportionate portion of shared multilingual vocabularies is allocated to high-frequency units from resource-rich languages. This imbalance is further compounded by the distinct Tamil script, re-

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 52–61
July 3, 2026 ©2026 Association for Computational Linguistics

sulting in minimal orthographic overlap with other languages. As illustrated in Figure 1, morphologically rich languages such as Tamil are often oversegmented, causing longer sequences with reduced morphological coherence leading to higher computational cost (Ahia et al., 2023; Petrov et al., 2023).

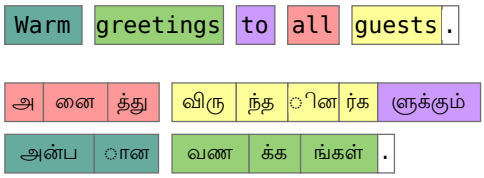


Figure 1: Example of over-fragmentation of a Tamil sentence by the Gemma tokenizer (Gemma Team, 2025). Words (visualized as boxes) are split into multiple subword tokens and token splits (expressed by |) do not align with morphological boundaries. For the English translation, the same tokenizer keeps all words as a single token.

Recent work demonstrates that linguistically-informed segmentation can improve morphological alignment and downstream performance in morphologically rich languages. Approaches integrating morphological supervision (Pan et al., 2020), rule-based constraints (Asgari et al., 2025), morphological analyzers (Matthews et al., 2018), architecture-level morphology modeling (Kumar M et al., 2010; Ataman and Federico, 2018), or morphology-aware pre-tokenization (Brahma et al., 2025) have shown gains in preserving morpheme boundaries and reducing over-segmentation. Yet, progress for Tamil remains limited, largely due to lacking large-scale high-quality morphological segmentation resources for tokenizer training and evaluation.

We introduce `TamilMorph`, a large-scale morphological segmentation dataset that contains over 480,000 valid morphological word forms with annotated morpheme boundaries to address the lack of high-quality resources for Tamil. Based on this gold standard, we present our morphology-aware tokenization framework `TamilTok`. By decomposing surface forms into valid morpheme sequences prior to subword learning, we improve morphological coherence and downstream performance on named entity recognition while maintaining competitive compression efficiency.

Our dataset and tokenizer are publicly available to support further research on Tamil NLP[1].

[1] https://gitlab.gwdg.de/huds/projects/tamiltok

2 Related Work

Tokenization is commonly conducted using frequency-driven subword segmentation methods, but their suitability for morphologically rich languages has been questioned. While prior work has demonstrated the potential benefits of linguistically-informed segmentation, progress remains uneven due to limited large-scale, task-aligned morphological datasets for many languages.

2.1 Subword Tokenization

Subword tokenization splits text input into sequences of frequently co-occurring characters. In contrast to word-level segmentation, it can capture novel sequences that were not seen during training (by fragmenting them into smaller parts) while maintaining a manageable vocabulary size. The most widely adopted approaches include Byte Pair Encoding (BPE) (Sennrich et al., 2016), which greedily merges the most frequent adjacent symbols; WordPiece (Devlin et al., 2019), which incrementally builds a subword vocabulary using likelihood-based merges that improve corpus modeling; Unigram Language Modeling (UnigramLM) (Kudo, 2018), which formulates tokenization as a probabilistic segmentation problem and iteratively prunes subword units that minimally contribute to the likelihood of the model.

Frameworks such as SentencePiece (Kudo and Richardson, 2018) enable language-independent preprocessing on raw text without relying on whitespaces, while byte-level models such as ByT5 (Xue et al., 2022) operate directly on raw bytes. Despite architectural differences, all subword segmentation approaches are fundamentally frequency-driven and largely language-agnostic without considering typological variation in morphological structure, causing large cross-lingual differences in tokenization quality.

2.2 Tokenization of Morphologically Complex Languages

Tokenizing morphologically rich languages presents unique challenges that standard subword approaches often fail to address adequately. Tamil, in particular, has been identified as one of the most fragmented languages in multilingual tokenization regimes, causing increased computational cost and reduced semantic coherence (Ahia et al., 2023; Petrov et al., 2023).

To address these issues, several studies explore morphology-aware tokenization strategies. For example, Pan et al. (2020) incorporate morphological preprocessing into BPE for neural machine translation for Turkish and Uyghur, Ataman et al. (2019) propose hierarchical character-based decoding that explicitly preserves morpheme boundaries during translation, and Matthews et al. (2018) introduce morpheme-level embeddings.

More recent analyses examine the influence of improved morphological alignment on downstream task performance. Previous research finds positive correlations between morphology-aware representations and downstream task accuracies (Vemula et al., 2025), but others conclude that the relationship is task- and language-dependent (Arnett et al., 2025), or resolved by equalizing data quantity (Arnett and Bergen, 2025). However, their evaluation does not include Tamil or other Dravidian languages. In contrast, Brahma et al. (2025) find that morphology-driven pre-tokenization improves performance for Hindi and Marathi using neural segmentation combined with corpus-level heuristics. These results indicate that morphology-aware tokenization can yield tangible benefits when properly aligned with the linguistic structure.

3 The TamilMorph Dataset

Previous approaches for morphological segmentation of Tamil morphology use finite-state analyzers (Sarveswaran et al., 2021), rule-based segmentation (Krishnan and Ragavan, 2021), or unsupervised approaches (Virpioja et al., 2013), but the existing systems only capture a very small set of morphological phenomena and do not generalize robustly. In order to assess morphological boundaries on a larger scale while maintaining grammatical accuracy, we introduce `TamilMorph`: an extensive dataset specifically designed for identifying linguistically reliable morpheme boundaries. It provides broad coverage of morphological phenomena such as inflectional and derivational patterns of nouns and verbs (including phonologically-conditioned alternations) while adhering to grammatical constraints.

3.1 Dataset Construction

Figure 2 visualizes our hybrid construction strategy that combines linguistic rules with controlled computational generation, ensuring both linguistic validity and a broad morphological coverage. The process follows a reverse engineering approach: Tamil words are first synthetically generated according to derivation and inflection rules while keeping track of segmentation boundaries. Using this controlled procedure for manipulating root forms, we avoid relying on an imprecise morphological analyzer. Instead, we focus on extracting high-quality root forms and linguistically valid rules.

Root Extraction We extract root morphemes directly from the annotated Tamil corpus provided by the Tamil Virtual Academy (Tamil Virtual Academy, 2022). Since our framework requires explicit identification of base forms, only the root information associated with each POS-tagged word form is utilized. After normalization and deduplication of the 320,448 annotated word forms, we obtained 28,209 noun roots and 3,078 verb roots.

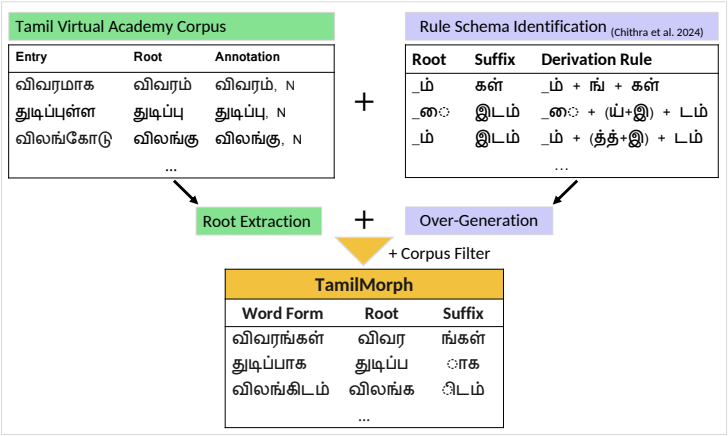


Figure 2: The dataset construction process for `TamilMorph`. We extract root morphemes and derivation rules from linguistic resources, over-generate all possible forms, and then filter based on corpus frequencies.

Rule Schema Identification We derive morphological rules from a grammar reference book for Tamil (Chithra et al., 2024), which documents the canonical patterns of well-formed derivation in Tamil, including sandhi processes governing phonological alternations at morpheme boundaries. As illustrated in Figure 2 in the *Formation Rule* column, the locative suffix இடம் surfaces differently depending on the phonological properties of the root ending ம் and ை. Distinct sandhi operations, indicated within parentheses, account for these variations, but the morphological function of the suffix remains the same.

Morphological Derivation Using the extracted root morphemes and linguistic rules, we implement a synthetic expansion procedure to overgenerate inflected and derived forms. For each root, all permissible suffix chains are applied according to morphological rules and phonological constraints. This process captures inflectional patterns for number, case, tense, mood, aspect, and voice for both nouns and verbs. In a second stage, the over-generated surface forms are checked against a strict corpus-based filter: only forms that occur at least once in the IndicNLP corpus (Kunchukuttan et al., 2020) are kept to avoid overly artificial generations. The dataset statistics are provided in Table 1.

Stage	Type	Count
Generation	Verbal suffixes	31
	Nominal suffixes	34
	Unique roots	31,287
	Generated forms	53,217,115
Filtered	**Word Forms**	**481,099**
	Unique root forms	101,821
	Unique suffixes	1,037
	Avg. suffix length	3.8

Table 1: Dataset statistics for `TamilMorph`.

3.2 TamilMorph

Using the procedure described above, we created `TamilMorph` to extract Tamil word forms with explicitly annotated morpheme boundaries that correspond to the correct linguistic structure. Each entry consists of a surface-form word decomposed into two components: a root and a corresponding suffix. To maintain structural consistency, all non-root morphemes associated with a word are combined into a single suffix unit, resulting in a fixed two-part representation (root + suffix) for every word form.[2] An example of this representation is shown in Figure 2.

The dataset systematically captures a broad spectrum of agglutinative inflectional and derivational Tamil patterns. Beyond serving as a linguistic resource, `TamilMorph` serves as the backbone of our `TamilTok` framework. The annotated segmentations enable supervised training of a neural morphological segmentation model, which generalizes beyond the dataset vocabulary and functions as an upstream component in the morphology-aware tokenization pipeline. Furthermore, the explicit morpheme annotations and broad lexical coverage provide a gold standard for evaluating tokenizer behavior using morphological alignment metrics such as MorphScore F_1 (Arnett and Bergen, 2025) and comparing tokenization strategies.

4 Experimental Setup

In our analysis, we compare the Tamil tokenization quality of newly trained monolingual tokenizers to a range of pretrained multilingual tokenizers: mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020), mT5 (Xue et al., 2021), Gemma (Gemma Team, 2025), mBART (Liu et al., 2020), IndicBERT (Doddapaneni et al., 2023). Our standardized evaluation setup compares the influence of the training corpus, the tokenization algorithm, and the vocabulary size.

4.1 Tokenizer Training

We train monolingual Tamil tokenizers using standardized normalization and preprocessing pipelines; the implementation details can be found in our code repository provided in the supplementary material. We use the three subword tokenization algorithms BPE (Sennrich et al., 2016), WordPiece (Devlin et al., 2019), and UnigramLM (Kudo, 2018), as implemented in the SentencePiece package.[3]

We use the Tamil portion of IndicNLP (Doddapaneni et al., 2023) with 68,237,343 lines and Tamil Wikipedia (Wikimedia Foundation, 2023) with 4,716,963 lines as training data. After filtering all lines containing non-Tamil characters, we retain 10,693,052 lines from Indic NLP and 2,997,493 from Wikipedia and ap-

[2] We found that the coarse-grained distinction provided more robust information for the tokenizer in pilot experiments, but we also provide the more fine-grained boundaries in our resource repository.

[3] `pypi.org/project/sentencepiece/0.2.0/`

ply Unicode normalization.[4] We vary the vocabulary size k of each tokenizer ($k \in \{3000, 5000, 8000, 16000, 32000\}$). Small values of k are comparable to the limited vocabulary allocation of Tamil in a shared multilingual vocabulary; larger values are more common for monolingual tokenizers.

4.2 Intrinsic Evaluation

We evaluate intrinsic tokenization quality along two complementary dimensions: information compression and linguistic plausibility. Information compression focuses on the efficiency of information reduction achieved by the tokenizer. In addition to the compression rate (ratio of characters to tokens), we analyze word-level fertility (average number of subword tokens per word) and Rényi efficiency (Zouhar et al., 2023), which measures how efficiently the tokenizer balances the token distribution over the vocabulary. Higher values for Rényi efficiency indicate that the usage of the vocabulary is closer to a uniform distribution. While this suggests more efficient resource allocation, it is counterintuitive from a linguistic perspective because in all natural languages, vocabulary usage follows a Zipfian distribution with a long tail of rare words. A highly efficient tokenizer does not necessarily use morphologically plausible boundaries to segment the input. Using our `TamilMorph` dataset, we can quantify the linguistic plausibility of the tokenizer by comparing subword splits to morpheme boundaries. We quantify the difference between the splits proposed by the tokenizer and the morphological boundaries between root and suffix using the Creutz–Linden F_1 (Morph-F_1) calculation (Creutz and Linden, 2004). Unlike the default MorphScore implementation (Arnett et al., 2025), we compute boundary statistics within a unified evaluation framework to accommodate format differences and allow consistent per-tokenizer aggregation. The resulting F_1-score penalizes both over- and under-segmentation, providing a direct and interpretable measure of how well a tokenizer preserves morphologically meaningful structure.

4.3 Downstream Performance

In order to quantify the interaction between intrinsic tokenization quality and downstream performance, we train a `TamilBERT` model using a standard encoder-only transformer architecture consisting of eight transformer blocks with eight attention heads. The model is pre-trained with a masked-language-modeling objective on the IndicNLP corpus using the different tokenizers and the smallest vocabulary size ($k = 3000$). The pre-trained TamilBERT models are finetuned on the WikiAnn Tamil NER dataset (Pan et al., 2017), which provides BIO-tagged annotations for the categories person (PER), location (LOC), and organization (ORG). The dataset contains 38,728 instances that are split into train (50%), dev (25%) and test (25%) sets. Finetuning is performed with a sequence length of 128 tokens, a batch size of 32, and a learning rate of 5×10^{-5} for 10 epochs.

5 TamilTok: Tokenization with Morphological Information

Subword tokenizers typically involve a pre-tokenization step that splits the input based on whitespaces. However, standard whitespace tokenization fails to capture internal morphological boundaries in agglutinative languages like Tamil. We aim at introducing a pre-tokenization step that splits the input based on morphological boundaries before learning subwords. Using our new dataset `TamilMorph`, we can now train a morphological segmentation model capable of decomposing unseen surface forms into valid morpheme sequences. We model segmentation as a conditional sequence generation task using ByT5 (Xue et al., 2022). We employ a byte-level architecture for this segmentation task as it robustly handles out-of-vocabulary characters without inducing prior biases, allowing the model to learn morphological boundaries directly from the provided data. We fine-tune the ByT5-Small configuration (300M parameters) on 503,292 root-suffix pairs in `TamilMorph` using a 90/10 train–test split without providing any auxiliary linguistic features or lexicon constraints. The model is trained for 5 epochs with a learning rate of 5×10^{-5} and deterministic decoding. The segmentation algorithm correctly identifies 96% of the boundaries between the root and the suffix on the test set.

If we apply our novel morphological segmentation algorithm as a pre-tokenization step, we force the subword algorithm to perform merge operations within morphologically valid units to reduce cross-morpheme fragmentation. The segmentation algorithm alone cannot be used directly as in-

[4] Unicode normalization ensures that visually similar Tamil characters are mapped to the same code points.

Lang	Type	Config	Fert.↓	Comp.	Rényi	Morph-F_1	NER-F_1
English	Multilingual	mBERT	1.32	4.61	.60	-	-
Tamil	Multilingual	mBERT	3.64	2.55	.47	.24	-
		XLM-R	2.47	3.79	.54	.31	-
		mT5	2.54	3.65	.55	.29	-
		Gemma	2.39	3.92	.56	.20	-
		mBART	2.47	3.79	.54	.31	-
		IndicBERT	3.39	2.73	.37	.11	-
	Monolingual	Byte-level BPE	2.51	3.73	**.87**	.26	.64
		Unigram	**2.27**	**3.96**	.85	.33	.65
		WordPiece	2.59	3.60	.84	.26	.65
	Morph. informed	`TamilTok`	2.50	3.75	.83	**.35**	**.67**

Table 2: Comparison of pre-trained multilingual tokenizers to our small monolingual tokenizers ($k = 3000$) and to our `TamilTok` variant with morphologically-informed pre-tokenization for the intrinsic evaluation metrics fertility, compression, and Rényi efficiency, for the alignment with morphological boundaries (Morph-F_1) and for downstream performance on named entity recognition (NER-F_1). For comparison, we additionally report the results for English using the mBERT tokenizer.

put for a neural model as it does not use a fixed vocabulary size and does not necessarily align with the surface structure of the input.

6 Results

In Table 2, we compare the tokenization quality of Tamil when using existing pretrained multilingual tokenizers to newly trained monolingual tokenizers.

6.1 Tokenizer Evaluation

For a fair comparison, we first provide the results for the monolingual tokenizers with the smallest vocabulary size ($k = 3000$). Fertility, compression ratio and Rényi efficiency are measured using parallel Tamil-English sentences from the Flores-101 dataset (Goyal et al., 2022), whereas the Morph-F_1 and NER-F_1 are measured using the `TamilMorph` test set and WikiAnn (Pan et al., 2017), respectively. The results confirm our anecdotal impression that existing pre-trained multilingual tokenizers perform poorly on Tamil inputs: high fertility and low compression rate indicate over-fragmentation that violates morphological boundaries. The mBERT model tokenizes English words into 1.32 subword tokens on average and Tamil words into 3.64 subword tokens. The monolingual tokenizers yield better information compression (especially the one based on the UnigramLM algorithm), and the high values for Rényi efficiency indicate a better allocation of the vocabulary. Our `TamilTok` shows a substantial improvement in morphological alignment without reducing information compression and downstream performance.

The choice of the training corpus exhibits only a marginal effect on alignment quality. For the UnigramLM algorithm, the tokenizers trained on the IndicNLP corpus achieve slightly higher fertility (3.91) and Morph-F_1 (0.33) compared to those trained on Wikipedia (3.87; 0.32). However, the differences are minimal, indicating that both corpora produce broadly comparable subword segmentation quality when trained under controlled and balanced conditions.

6.2 Tokenization Parameters

For the following experiments, we chose the IndicNLP corpus due to its larger overall token count and higher number of unique word forms to better examine the influence of the vocabulary size.

Tokenization Algorithm Figure 3 illustrates the influence of the vocabulary size and the tokenization algorithm on the F_1 and fertility metrics. Across all configurations, using UnigramLM consistently leads to better morphological alignment than BPE or WordPiece. Crucially, this advantage is not accompanied by substantially higher fertility, indicating that the gains cannot be attributed to increased subword fragmentation. Rather, the results suggest that the probabilistic objective underlying UnigramLM more effectively captures morphologically coherent boundaries than deterministic merge-based approaches.

Vocabulary Size Increasing the vocabulary systematically reduces fertility, reflecting improved compression efficiency through longer and more

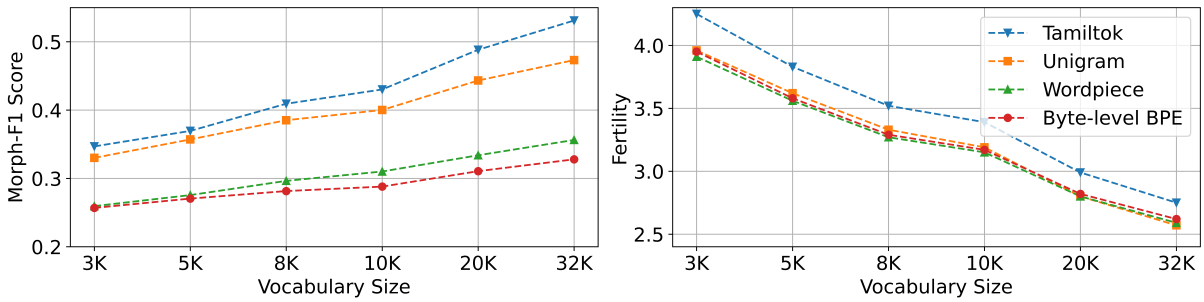


Figure 3: Effect of tokenization algorithm and vocabulary size on morphological alignment (Morph-F_1, left) and fertility (right). Both metrics improve with increasing vocabulary size and the UnigramLM algorithm consistently outperforms BPE and WordPiece in preserving morphological boundaries.

consolidated subword units. At the same time, morphological alignment increases steadily across the entire vocabulary range. The absence of saturation within the examined range indicates that morphological segmentation quality generally benefits strongly from significantly larger vocabulary sizes independent of the segmentation algorithm.

Evaluating fertility jointly with alignment therefore provides a more faithful characterization of tokenizer quality and clearly establishes the superiority of UnigramLM. Furthermore, as expected, larger vocabulary sizes offer increased capacity for extracting linguistically meaningful subword units, contributing to improved alignment alongside reduced fragmentation.

7 Discussion

We observe three main observations in our experiments. First, multilingual tokenizers exhibit substantial over-segmentation for Tamil, as reflected in high fertility and low morphological alignment scores. This confirms prior concerns (Ahia et al., 2023; Arnett and Bergen, 2025; Petrov et al., 2023) that frequency-driven shared vocabularies disproportionately disadvantage morphologically rich and non-Latin-script languages. For Tamil, the resulting fragmentation disrupts morpheme boundaries and increases sequence length.

Second, the tokenization algorithm and the vocabulary size have a stronger influence on morphological alignment than the choice of the training corpus. UnigramLM consistently outperforms BPE and WordPiece in Morph-F_1 across controlled settings and both compression and alignment benetfit from larger vocabulary sizes. Importantly, fertility alone does not sufficiently capture qualitative tokenization differences, underscoring the need for linguistically motivated evaluation.

Third, explicitly combining morphological information with existing frequency-based tokenization algorithms in `TamilTok` further improves morphological alignment, in line with previous work (Brahma et al., 2025; Bayram et al., 2025; Jabbar, 2023; Krishnan and Ragavan, 2021). Although morphology-aware preprocessing introduces a slight increase in fertility, the gain in Morph-F_1 indicates better preservation of linguistically meaningful units. This suggests that minor compression trade-offs may be justified when structural coherence is improved.

The proposed `TamilMorph` dataset plays a central role in enabling these analyses. Its rule-derived boundary-level annotations provide a consistent gold standard for evaluating tokenizer–morpheme agreement and support supervised training of neural segmentation models. It can be used for training and benchmarking in future research on Tamil morphological segmentation and tokenization.

8 Conclusion

Our work introduces two contributions to morphology-aware NLP for Tamil. We present `TamilMorph`, a large-scale morphological segmentation dataset comprising more than 480,000 linguistically validated word forms with explicit morpheme segmentations. The dataset offers a reproducible gold standard for evaluating morphological alignment and supports supervised morphological segmentation.

The proposed `TamilTok` algorithm, a morphology-aware tokenization framework, integrates neural morphological segmentation into subword training. By injecting morpheme structure prior to tokenizer training, the framework improves morphological alignment while maintaining competitive compression efficiency

and downstream performance.

Our approach can be extended to other agglutinative languages, provided suitable morphological resources are available. Overall, our work demonstrates that linguistically-informed tokenization leads to more structurally coherent representations and offers a practical pathway toward improved modeling of morphologically rich languages.

Limitations

We extracted the root forms from the Tamil Virtual Academy corpus (Tamil Virtual Academy, 2022) without systematically ensuring their lexical minimality. Manual examination indicates that some entries labeled as roots in the corpus correspond to compound forms (that function as surface-level bases) rather than irreducible morphological units. We do not expect this limitation to affect the segmentation performance or structural consistency of the dataset, but it may influence the strict linguistic validity of the root annotations and thus, to an extent, affect the representation quality of linguistic stems on downstream tasks.

The derivational rules used for generating the forms in the dataset are not exhaustive. The full range of possible derivational suffixes cannot be derived from a single corpus or limited word-formation references. Expanding the inventory of suffixes will require broader linguistic investigation and the integration of additional lexical resources.

Our corpus-based filter for ensuring lexical validity is overly strict. It might exclude forms that occur with high frequency in domains not covered in the IndicNLP corpus.

Acknowledgments

This research is partially supported by the zukunft.niedersachsen program of the VolkswagenStiftung (LB) and by a VENI grant (Vl.Veni.211C.039) from the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (AH, LB).

References

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David Mortensen, Noah Smith, and Yulia Tsvetkov. 2023. Do all languages cost the same? tokenization in the era of commercial language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9904–9923, Singapore. Association for Computational Linguistics.

Catherine Arnett and Benjamin Bergen. 2025. Why do language models perform worse for morphologically complex languages? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6607–6623. Association for Computational Linguistics.

Catherine Arnett, Marisa Hudspeth, and Brendan O'Connor. 2025. Evaluating morphological alignment of tokenizers in 70 languages. In *Proceedings of the ICML 2025 Tokenization Workshop (TokShop)*.

Ehsaneddin Asgari, Yassine El Kheir, and Mohammad Ali Sadraei Javaheri. 2025. MorphBPE: A morpho-aware tokenizer bridging linguistic complexity for efficient LLM training across morphologies. *Preprint*, arXiv:2502.00894.

Duygu Ataman and Marcello Federico. 2018. Compositional representation of morphologically-rich input for neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 305–311, Melbourne, Australia. Association for Computational Linguistics.

Duygu Ataman, Orhan Firat, Mattia A. Di Gangi, Marcello Federico, and Alexandra Birch. 2019. On the importance of word boundaries in character-level neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 187–193, Hong Kong. Association for Computational Linguistics.

M Ali Bayram, Ali Arda Fincan, Ahmet Semih Gümüş, Sercan Karakaş, Banu Diri, Savaş Yıldırım, and Demircan Çelik. 2025. Tokens with meaning: A hybrid tokenization approach for nlp. *arXiv preprint arXiv:2508.14292*.

Maharaj Brahma, NJ Karthika, Atul Singh, Devaraj Adiga, Smruti Bhate, Ganesh Ramakrishnan, Rohit Saluja, and Maunendra Sankar Desarkar. 2025. Morphtok: Morphologically grounded tokenization for Indian languages. *arXiv e-prints*, pages arXiv–2504.

Deepthi Chintha and Nanda Vikas Konduru. 2025. Survey of tokenization mechanisms in multilingual large language models with a focus on Indian languages. *Journal of Emerging Technologies and Innovative Research*.

V Chithra, V Balamurugan, N Rajendiran, M Tamilarasan, and P Kavitha. 2024. Tamizhai pizhaiyindri ezhudhivom. https://www.tamilvu.org/sites/default/files/notice/Thamizhai_Pizhaiyinri_Ezhuthuvom.pdf.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. 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.

Mathias Johan Philip Creutz and Bo Krister Johan Linden. 2004. Morpheme segmentation gold standards for Finnish and English. In *Publications in Computer and Information Science: Report A77*. Helsinki University of Technology.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2023. Towards leaving no Indic language behind: Building monolingual corpora, benchmark and models for Indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12402–12426, Toronto, Canada.

Gemma Team. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Haris Jabbar. 2023. Morphpiece: A linguistic tokenizer for large language models. *arXiv preprint arXiv:2307.07262*.

Arjun Sai Krishnan and Seyoon Ragavan. 2021. Morphology-aware meta-embeddings for Tamil. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 94–111, Online. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71. Association for Computational Linguistics.

Anand Kumar M, Dhanalakshmi V, Soman K.P, and Rajendran S. 2010. A novel data driven algorithm for Tamil morphological generator. *International Journal of Computer Applications*, 6(12):52–56.

Anoop Kunchukuttan, Divyanshu Kakwani, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. AI4Bharat-IndicNLP Corpus: Monolingual corpora and word embeddings for Indic languages. *arXiv preprint arXiv:2005.00085*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Austin Matthews, Graham Neubig, and Chris Dyer. 2018. Using morphological knowledge in open-vocabulary neural language models. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1435–1445.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Yirong Pan, Xiao Li, Yating Yang, and Rui Dong. 2020. Morphological word segmentation on agglutinative languages for neural machine translation. *Preprint*, arXiv:2001.01589.

Iñigo Parra. 2024. Morphological typology in BPE subword productivity and language modeling. In *Latinx in AI@ NeurIPS 2024*.

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. In *Advances in Neural Information Processing Systems*, volume 36, pages 36963–36990. Curran Associates, Inc.

Kengatharaiyer Sarveswaran, Gihan Dias, and Miriam Butt. 2021. Thamizhi Morph: A morphological parser for the Tamil language. *Machine Translation*, 35(1):37–70.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Jothir Adithya T K, Nithish Kumar S, Felicia Lilian J, and Mahalakshmi S. 2024. Monolingual text summarization for Indic languages using LLMs. In *Proceedings of the 21st International Conference on Natural Language Processing (ICON)*, pages 94–101, AU-KBC Research Centre, Chennai, India. NLP Association of India (NLPAI).

Tamil Virtual Academy. 2022. Tamil annotated corpus. https://github.com/Tamil-Virtual-Academy/Tamil-Annotated-Corpus. Accessed: 2026-02-25.

Saketh Reddy Vemula, Sandipan Dandapat, Dipti Sharma, and Parameswari Krishnamurthy. 2025. Rethinking tokenization for rich morphology: The dominance of unigram over BPE and morphological alignment. In *The 14th International Joint Conference on Natural Language Processing and The 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 232–252, Mumbai, India. Association for Computational Linguistics.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline. Aalto University publication series SCIENCE + TECHNOLOGY, 25/2013.

Wikimedia Foundation. 2023. Wikimedia downloads. Accessed: 10-10-2025.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Vilém Zouhar, Clara Meister, Juan Gastaldi, Li Du, Mrinmaya Sachan, and Ryan Cotterell. 2023. Tokenization and the noiseless channel. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5184–5207, Toronto, Canada. Association for Computational Linguistics.

Thiruppugazh-KG Dataset: A Manually Annotated Resource for Computational Analysis of Tamil Devotional Literature

Garthigan Kumarasamy, Jubeerathan Thevakumar, Sathurgini Uthayakumar, Disne Kajanath, Narthana Sivalingam, Uthayasanker Thayasivam
Department of Computer Science and Engineering,
University of Moratuwa, Sri Lanka
{garthigank.25,jubeerathan.25,sathurgini.25}
{disne.25,narthanas,rtuthaya}@cse.mrt.ac.lk

Abstract

This paper introduces Thiruppugazh-KG, a semantically annotated dataset and knowledge graph derived from the Thiruppugazh corpus, a 14th-century collection of 1,335 Tamil devotional hymns composed by Arunagirinathar. The dataset includes annotations for entities, devotional themes, mythological events, philosophical concepts, imagery, and sacred locations mentioned in each hymn. Using these annotations, we construct a Neo4j-based knowledge graph that models relationships between hymns and their associated cultural and narrative elements. Graph analytics, including PageRank, are applied to identify prominent entities and sacred locations within the corpus. The resulting resource provides a structured representation of Tamil devotional literature and supports computational analysis of cultural texts in low-resource languages.

1 Introduction

Tamil devotional literature forms a significant part of South Asian religious and literary culture. Thiruppugazh[1], composed by the 14th-century poet-saint Arunagirinathar, is one of the most celebrated devotional works in Tamil tradition. The corpus contains 1,335 hymns dedicated primarily to Lord Murugan and is renowned for its intricate poetic style, complex rhythmic structure, and vivid devotional symbolism (Zvelebil, 1973) . These hymns combine mythology, spiritual philosophy, and descriptions of sacred temples, creating a rich literary and religious landscape that has been studied extensively within Tamil scholarship.

Recent developments in digital humanities have introduced computational approaches for analyzing large literary corpora. Methods such as distant reading and macroanalysis allow scholars to identify large-scale thematic and structural patterns across texts rather than focusing only on individual works (Moretti, 2013) (Jockers, 2013). Within this context, knowledge graphs provide a powerful framework for representing semantic relationships extracted from textual data. In a knowledge graph, entities and concepts are modeled as nodes while relationships between them are represented as edges, enabling network analysis, semantic querying, and discovery of hidden patterns within complex datasets. Knowledge graphs model entities and relationships as nodes and edges, enabling network analysis and semantic querying of complex datasets (Hogan et al., 2021) (Robinson et al., 2015).

Knowledge graph techniques have been successfully applied in domains such as web knowledge bases, digital libraries, and cultural heritage datasets. Projects such as DBpedia demonstrate how structured knowledge can be extracted from textual sources and organized into machine-readable semantic networks (Auer et al., 2007) (Bizer et al., 2023). Despite these advances, similar computational approaches remain relatively rare in the study of Tamil devotional literature and Bhakti poetry.

This study addresses this gap by representing the Thiruppugazh corpus as a knowledge graph. A manually annotated dataset of 1,335 songs was created containing entities, mythological events, devotional themes, philosophical concepts, symbolic imagery, and temple locations mentioned in each hymn. Using these annotations, a Neo4j-based knowledge graph was constructed and analyzed using graph algorithms to identify dominant themes, frequently occurring mythological narratives, and prominent temples referenced within the corpus.

The main contributions of this work are as follows:

1. We introduce Thiruppugazh-KG, the first semantically annotated knowledge graph dataset

[1]https://en.wikipedia.org/wiki/Tiruppukal

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 62–70
July 3, 2026 ©2026 Association for Computational Linguistics

for Tamil devotional literature.

2. We provide manual annotations for 1,335 hymns covering entities, themes, mythology, philosophy, imagery, and sacred geography.

3. We construct a Neo4j-based devotional knowledge graph supporting semantic querying and graph analytics.

4. We demonstrate the utility of the resource through graph-based retrieval and computational literary analysis.

2 Related Work

Computational approaches in digital humanities have enabled new ways of analyzing large literary corpora using semantic and data-driven methods. Techniques such as knowledge graphs, network analysis, and probabilistic text modeling help represent and analyze relationships among entities in cultural and literary datasets. This section reviews previous research related to knowledge graphs and computational approaches used in literary analysis.

2.1 Knowledge Graphs

Knowledge graphs are structured representations of entities and relationships that enable semantic reasoning and data integration across complex domains. Hogan et al. define knowledge graphs as networks of real-world entities connected by semantic relationships that enable knowledge discovery and reasoning across heterogeneous data sources (Hogan et al., 2021). Such graphs have become fundamental infrastructure in modern data management and artificial intelligence systems.

Large-scale knowledge graphs such as DBpedia extract structured information from Wikipedia and represent it in RDF format, enabling semantic web applications and knowledge discovery (Auer et al., 2007). These systems demonstrate the potential of graph-based representations for organizing large bodies of information. Graph databases such as Neo4j provide efficient implementations of the property graph model and support complex graph queries and algorithms. Robinson, Webber, and Eifrem note that graph databases are particularly well suited for domains involving highly connected data structures (Robinson et al., 2015).

2.2 Knowledge Graphs in Digital Humanities

Digital humanities research increasingly uses knowledge graphs to represent cultural heritage data, historical texts, and literary networks. Romanello and colleagues demonstrate that knowledge graphs allow scholars to represent relationships among texts, authors, places, and historical events in a machine-readable structure that supports computational analysis (Kokash et al., 2023).

Knowledge graph models have also been applied to historical archives, museum collections, and literary corpora to capture relationships between characters, events, and locations. These approaches enable researchers to analyze narrative structures and cultural networks using computational techniques (Crane et al., 2006; McCarty, 2005).

2.3 Computational Literary Analysis

Computational literary studies use data-driven techniques to analyze narrative structures, themes, and character networks in literary texts. Newman's work on network science provides theoretical foundations for analyzing relationships within complex networks, including social and narrative networks (Newman, 2001).

Network analysis has been applied to classical literature, including Shakespeare's plays and epic narratives, to examine relationships among characters and thematic patterns. Similarly, distant reading approaches proposed by Moretti emphasize analyzing large collections of texts using quantitative methods rather than focusing solely on individual works (Moretti, 2013).

Topic modeling techniques and probabilistic text models have also been used to identify thematic structures within literary corpora (Blei, 2012). These computational methods demonstrate the value of large-scale textual analysis for literary scholarship.

2.4 Tamil Literature Studies

Tamil literature has been extensively studied by scholars such as Zvelebil, who documented its historical development and cultural significance, particularly within the context of devotional poetry and religious traditions (Zvelebil, 1973). Murugan devotion occupies a central place in Tamil religious literature and continues to influence contemporary cultural practices.

However, computational approaches to Tamil literary analysis remain relatively limited. Most existing research focuses on linguistic processing, digitization, and corpus creation rather than semantic knowledge modeling. The present study contributes to this emerging field by introducing

a knowledge graph representation for Tamil devotional literature.

3 Dataset Description

The dataset used in this study is derived from the *Thiruppugazh* corpus, a classical Tamil devotional literary work composed by the 14th-century saint-poet Arunagirinathar. The corpus contains 1,335 devotional hymns dedicated primarily to Lord Murugan, a central deity in Tamil Hindu spiritual traditions. The hymns are known for their complex poetic structure, mythological references, symbolic imagery, and philosophical teachings.

To enable computational analysis, the Thiruppugazh corpus was transformed into a structured dataset with semantic annotations capturing mythological entities, devotional themes, philosophical ideas, and narrative events. This structured representation supports tasks such as knowledge extraction, semantic search, and digital humanities analysis.

3.1 Data Source

The original Tamil lyrics and explanatory interpretations were collected from the Kaumaram Thiruppugazh digital archive[2], which preserves the works of Arunagirinathar. Digital repositories of classical literature are increasingly used in digital humanities research as they enable computational analysis of historical texts and cultural narratives (Crane et al., 2006).

Each hymn entry contains the original Tamil text, Tamil explanatory notes, and an English interpretation. Additional semantic annotations were created to capture key literary and mythological concepts present in the hymns.

3.2 Dataset Structure

Each hymn is represented as a structured record containing linguistic content and semantic annotations. These fields allow systematic analysis of devotional themes, mythological narratives, and philosophical concepts present in the Thiruppugazh corpus. The dataset schema is summarized in Table 1.The final dataset contains semantic annotations for all hymns in the Thiruppugazh corpus. Each hymn has been structured to capture both linguistic information and semantic metadata.The overall dataset statistics are summarized in Table 2.

[2]https://www.kaumaram.com/thiru/

Field Name	Description
song_number	Unique identifier assigned to each hymn.
song_name	Title or commonly recognized name of the hymn.
abode	Sacred location associated with Murugan referenced in the hymn.
song_tamil	Original Tamil lyrics of the hymn.
definition_tamil	Explanatory interpretation of the hymn in Tamil.
definition_english	English explanation describing the meaning of the hymn.
themes	Major devotional or spiritual themes present in the hymn.
entities	Named entities such as deities, mythological characters, or places.
myth_events	Mythological narratives or divine events described in the hymn.
philosophy	Spiritual or philosophical concepts conveyed in the text.
imagery	Symbolic or poetic imagery used within the hymn.
relationships	Interactions between divine figures or characters.
summary	Concise description summarizing the hymn.
keywords	Key terms representing major concepts in the hymn.
languages	Languages available in the record (Tamil and English).

Table 1: Schema of the Thiruppugazh dataset

The annotated Thiruppugazh-KG dataset used in this study is publicly available to support reproducibility and future research in Tamil digital humanities and low-resource NLP. The repository[3] includes the structured JSON dataset, annotation guidelines, Neo4j knowledge graph resources, example Cypher queries, and documentation for dataset construction and usage.

3.3 Annotation Methodology

The dataset annotations were created through a structured manual annotation process carried out

[3]https://github.com/Garthigan/Thiruppugazh-KG-Dataset

Attribute	Value
Corpus Name	Thiruppugazh
Author	Arunagirinathar
Number of Hymns	1,335
Languages	Tamil , English
Annotation Types	Themes, Entities, Myth Events, Imagery, Philosophy, Relationships
Data Source	Kaumaram Thiruppugazh Archive
Data Format	JSON structured records
Domain	Tamil devotional literature

Table 2: Statistics of the Thiruppugazh dataset

by two annotators with background in Tamil literature and computational linguistics. Both annotators were familiar with the cultural and religious context of the Thiruppugazh corpus. In cases where disagreements occurred, a third reviewer with expertise in Tamil devotional literature was consulted to resolve conflicts.

Each hymn in the Thiruppugazh corpus, along with its accompanying interpretation from the Kaumaram digital archive, was carefully examined to identify important semantic components. The annotation focused on extracting key elements such as named entities, devotional themes, mythological events, philosophical concepts, and poetic imagery.

The annotation process followed a predefined workflow. First, the Tamil hymn and its interpretation were reviewed to understand the narrative and symbolic meaning. Next, semantic elements such as divine figures, mythological references, and spiritual teachings were identified. These elements were then categorized into predefined annotation fields including *themes*, *entities*, *imagery*, *mythological events*, *philosophical concepts*, and *relationships*.

To ensure consistency across annotations, a set of annotation guidelines was developed describing the definition and scope of each category. For example, *entities* correspond to divine figures or mythological characters referenced in the hymn, while *themes* represent the primary devotional or spiritual ideas expressed in the verse. *Mythological events* capture references to well-known narratives from Hindu mythology, and *imagery* represents symbolic metaphors used in the poetic language of the hymn.

To evaluate annotation consistency, Whole hymns was independently annotated by both annotators. Inter-annotator agreement was measured using Cohen's Kappa (Cohen, 1960) coefficient across the major annotation categories. The resulting agreement score was $\kappa = 0.78$, indicating substantial agreement between annotators.

This annotation approach enables a structured representation of literary knowledge while preserving the religious and cultural context of the original hymns. The resulting dataset supports computational analysis of Tamil devotional literature and facilitates tasks such as semantic search, knowledge graph construction, and knowledge-based retrieval for large language models.

The overall workflow used to construct the dataset is illustrated in Figure 1. The process begins with collecting the hymns and interpretations from the Kaumaram digital archive, followed by manual textual analysis and semantic annotation. The annotated information is then organized into a structured JSON format to create the final Thiruppugazh dataset.

4 Knowledge Graph Construction

The knowledge graph for the Thiruppugazh corpus was constructed using the Neo4j graph database, which is a widely adopted platform for implementing property graph–based knowledge systems (Robinson et al., 2015). Neo4j enables efficient storage and querying of complex relationships among entities, making it suitable for representing the semantic structure of devotional literature(Neo4j Inc., 2023). In this work, the knowledge graph models relationships between songs, mythological references, philosophical ideas, and geographic locations mentioned in the Thiruppugazh text.

4.1 Node Relationship Types

The graph schema consists of several node types representing the key semantic components extracted from the Thiruppugazh corpus. Each node represents a distinct concept or entity within the text. Those are Song, Entity, Theme, MythEvent, Philosophy, Imagery and Place. These node types enable structured representation of the literary and spiritual knowledge embedded within the Thiruppugazh corpus.

The semantic connections between nodes are modeled through directed relationships. These re-

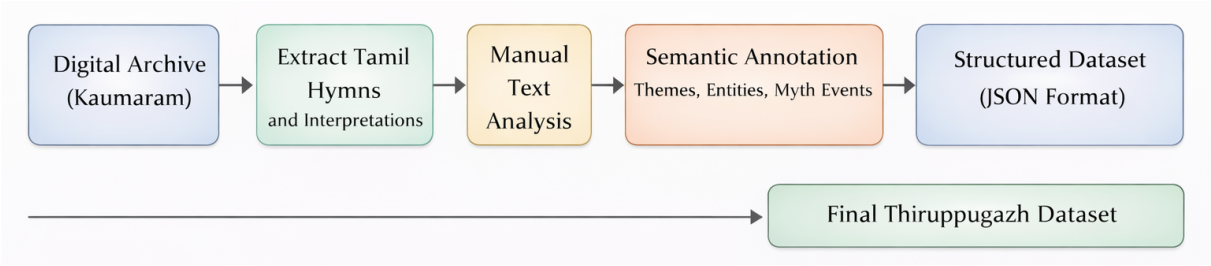


Figure 1: Annotation workflow used to construct the Thiruppugazh dataset

Figure 2: Architecture of the Thiruppugazh Knowledge Graph showing the relationships between Song nodes

Figure 3: Example subgraph for Song 1183 "Porudha kayalvizhi"

lationships capture the contextual associations between songs and their corresponding thematic or mythological elements.

The overall structure of these node types and relationships is illustrated in Figure 2, which presents the conceptual architecture of the Thiruppugazh knowledge graph.

4.2 Example Subgraph

An example of a subgraph representing a Thiruppugazh hymn is illustrated below Figure 3

This representation captures the semantic relationships between the hymn, the deity Murugan, the mythological event referenced in the text, and the sacred location associated with the song.

4.3 Murugan Devotional Network

To further analyze the devotional structure of the Thiruppugazh corpus, a Murugan-centered subgraph was extracted from the knowledge graph. This subgraph highlights how multiple hymns are connected through references to the deity Murugan and their associated thematic and mythological elements.

The following Cypher query was used to retrieve songs that mention Murugan and their related semantic nodes:

Listing 1: Cypher query used to extract the Murugan devotional network

```
MATCH (e:Entity {name:"Murugan"})<-[:
    MENTIONS_ENTITY]-(s:Song)
MATCH (s)-[r]->(n)
RETURN e, s, r, n
LIMIT 100
```

The resulting network places Murugan as the central entity, connected to multiple Song nodes through the MENTIONS_ENTITY relationship. Each song node is further linked to additional semantic nodes such as Theme, MythEvent, and Place, which represent the thematic context, mythological references, and sacred locations associated with the hymn.

As illustrated in Figure 4, this structure forms a devotional knowledge network where Murugan acts as the central hub connecting various hymns and their corresponding semantic attributes within the Thiruppugazh corpus.

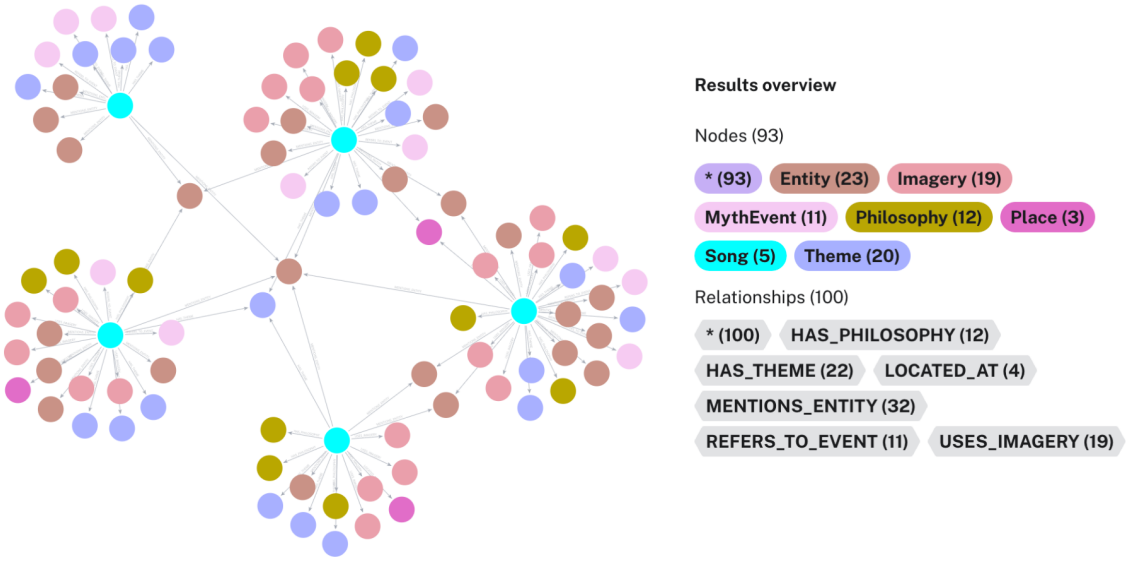


Figure 4: Murugan-centered devotional network extracted from the Thiruppugazh knowledge graph

5 Graph Analytics

To provide an overview of the constructed knowledge graph, Table 3 summarizes key structural statistics including the number of nodes, edges, average degree, and detected communities.

Metric	Value
Nodes	11992
Edges	28228
Average Degree	4.70
Communities	9758

Table 3: Summary statistics of the Thiruppugazh knowledge graph

Graph analytics techniques were applied to the constructed knowledge graph to explore the structural and semantic relationships present within the Thiruppugazh corpus. By leveraging the graph-based representation implemented in Neo4j, it becomes possible to analyze how devotional themes, mythological events, and sacred locations are interconnected through individual hymns.

The PageRank algorithm was applied to identify the most central entities within the Thiruppugazh knowledge graph.The computation was performed using the Neo4j Graph Data Science (GDS) library with a damping factor of 0.85.To facilitate interpretation and comparison, the resulting PageRank scores were normalized using Min–Max scaling.

Entity	PageRank Score
Murugan	1.0
Siva	0.32
Valli	0.30
Brahma	0.19

Table 4: Top entities ranked by normalized PageRank scores in the Thiruppugazh knowledge graph

Table 4 presents the top ranked entities based on the normalized PageRank scores. While Murugan naturally appears as the highest ranked entity due to the devotional focus of the corpus, the relative prominence of other entities provides additional insight into the structure of the narrative network. Figures such as Siva and Valli appear as highly influential nodes, reflecting their strong mythological and relational connections within the hymns. Brahma also emerges as a notable entity, indicating the presence of broader Hindu cosmological references within the corpus.

These rankings highlight how the knowledge graph captures relationships among divine figures and mythological elements across the hymns. In particular, the prominence of entities such as Siva and Valli suggests recurring narrative associations with Murugan that contribute to the interconnected structure of the devotional network.

Community detection analysis was further performed to examine the structural organization of

the devotional network. The detected communities revealed clusters corresponding to battle mythology, sacred geography, and devotional philosophy. These clusters indicate that semantically related hymns form densely connected subgraphs within the knowledge graph, demonstrating the effectiveness of the proposed representation in capturing thematic and narrative relationships across the corpus.

6 Results and Discussion

The constructed knowledge graph enables quantitative analysis of the Thiruppugazh corpus by examining the frequency of themes, mythological events, sacred locations, and relationship types extracted from the hymns. Through graph-based representation and querying, the dataset allows systematic exploration of devotional patterns and narrative structures embedded within the corpus.

The analysis of sacred places shows that several temple locations frequently appear in the hymns. Among these, Pazhani (98 occurrences), Thiruchchendhur (83), Thiruvaruṇai (81), Chidhambaram (67), and Thiruththanigai (64) are prominently referenced. These locations correspond to historically significant Murugan temples and pilgrimage centers in Tamil religious tradition. The frequency of these places within the graph highlights how devotional poetry is strongly associated with sacred geography, where hymns not only express spiritual devotion but also situate that devotion within specific temple sites and pilgrimage contexts.

The analysis of mythological references reveals recurring narrative motifs across the corpus. Several events associated with Murugan's battles against demonic forces appear frequently, including Murugan defeating Suran (45 occurrences), Murugan's victory over demons (41), and destruction of Suran (33). Other mythological narratives such as Murugan's marriage to Vaḷḷi, the destruction of Thiripuram, and the churning of the ocean are also represented in the graph. These results demonstrate that Thiruppugazh hymns incorporate a wide range of mythological episodes drawn from broader Hindu narrative traditions, which are interwoven with devotional expression.

Thematic analysis further reveals the distribution of spiritual concepts within the corpus. The theme devotion to Murugan appears most frequently (969 occurrences), followed by spiritual longing (234), spiritual enlightenment (168), divine intervention (164), spiritual awakening (149), and divine protection (131). While the devotional focus of the corpus is already recognized in traditional literary studies, the knowledge graph provides a structured and quantitative representation of these themes, enabling computational examination of how devotional ideas recur and connect across multiple hymns.

The distribution of relationship types within the graph also provides insights into the structural composition of the dataset. The most common relationship is MENTIONS_ENTITY (8,564 occurrences), indicating that references to divine figures and mythological characters form a central component of the corpus. This is followed by USES_IMAGERY (6,166), HAS_THEME (5,544), and HAS_PHILOSOPHY (4,104), which highlight the extensive use of poetic imagery, thematic expression, and philosophical reflection in the hymns. Relationships such as REFERS_TO_EVENT (2,515) and LOCATED_AT (1,335) further link songs to mythological narratives and sacred places. Together, these structural patterns demonstrate how the knowledge graph captures multiple layers of semantic information, including narrative, symbolic, and geographical dimensions of the Thiruppugazh corpus.

Table 5 summarizes the two most frequent occurrences in each category extracted from the knowledge graph. These results provide a compact overview of the most prominent themes, mythological narratives, sacred locations, and relationship types represented in the graph.

6.1 Limitations and Future Work

Although the proposed knowledge graph successfully represents important semantic relationships within the Thiruppugazh corpus, several limitations remain.

First, the current system relies on manual or semi-automatic extraction of entities and themes, which may limit scalability when processing larger textual datasets. Automated natural language processing techniques could be incorporated in future work to improve the extraction of entities, relationships, and semantic concepts from the corpus.

Second, the present graph focuses primarily on a limited set of node types such as songs, themes, entities, mythological events, and places. Future research could extend the schema to include additional semantic layers such as poetic structure, lin-

Category	Rank 1 (Frequency)	Rank 2 (Frequency)
Place	Pazhani (98)	Thiruchchendhur (83)
Mythological Event	Murugan defeating Suran (45)	Murugan's victory over demons (41)
Theme	Devotion to Murugan (969)	Spiritual longing (234)
Relationship Type	MENTIONS_ENTITY (8564)	USES_IMAGERY (6166)

Table 5: Top Two Frequent Concepts Extracted from the Thiruppugazh Knowledge Graph

guistic features, historical context, and inter-textual references.

Future work may also integrate advanced graph analytics techniques, including community detection, centrality measures, and knowledge graph embeddings, to further analyze the devotional structure and thematic relationships present in the corpus. Additionally, interactive visualization tools could be developed to allow scholars and researchers to explore the knowledge graph more effectively.

7 Conclusion

We present a manually annotated knowledge graph dataset derived from the Thiruppugazh corpus to model semantic relationships between songs, themes, mythological events, entities, imagery, philosophical concepts, and sacred locations. The annotations were used to construct a Neo4j-based knowledge graph that enables structured exploration of devotional literature. Our analysis shows that devotion to Murugan is the most dominant theme, while sacred locations such as Pazhani and Thiruchchendhūr frequently appear in the hymns. These findings demonstrate that knowledge graph–based representations can effectively capture the semantic structure of the Thiruppugazh corpus. This work provides a foundation for future computational analysis and knowledge-driven exploration of Tamil devotional literature.

Acknowledgments

The author sincerely thank the volunteers who assisted in manually annotating the Thiruppugazh dataset.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *international semantic web conference*, pages 722–735. Springer.

Christian Bizer, Tom Heath, and Tim Berners-Lee. 2023. Linked data-the story so far. In *Linking the World's Information: Essays on Tim Berners-Lee's Invention of the World Wide Web*, pages 115–143.

David M Blei. 2012. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Gregory Crane, David Bamman, Lisa Cerrato, Alison Jones, David Mimno, Adrian Packel, David Sculley, and Gabriel Weaver. 2006. Beyond digital incunabula: Modeling the next generation of digital libraries. In *International Conference on Theory and Practice of Digital Libraries*, pages 353–366. Springer.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, and 1 others. 2021. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37.

Matthew L Jockers. 2013. *Macroanalysis: Digital methods and literary history*. University of Illinois Press.

Natallia Kokash, Matteo Romanello, Ernest Suyver, and Giovanni Colavizza. 2023. From books to knowledge graphs. *Journal of Data Mining & Digital Humanities*, 2023.

Willard McCarty. 2005. *Humanities computing*. Springer.

Franco Moretti. 2013. *Distant reading*. Verso Books.

Neo4j Inc. 2023. *Neo4j Graph Data Science Documentation*. Accessed: 2026-03-01.

Mark EJ Newman. 2001. The structure of scientific collaboration networks. *Proceedings of the national academy of sciences*, 98(2):404–409.

Ian Robinson, Jim Webber, and Emil Eifrem. 2015. *Graph databases: new opportunities for connected data*. " O'Reilly Media, Inc.".

Kamil Zvelebil. 1973. *The Smile of Murugan: On Tamil Literature of South India*. Brill.

A Thiruppugazh-KG Dataset Example

Examples in Thiruppugazh-KG Dataset are shown in Table 6.

Field	Example Value
song_number	0001
song_name	Muththaiththaru
abode	thiruvarunai
entities	Murugan, Siva, Brahma, Vishnu, Ravana, Arjuna
themes	devotion to Murugan; divine protection; spiritual teachings
myth_events	preaching of OM; destruction of Ravana's heads; churning of the ocean
philosophy	Karma Yoga; spiritual enlightenment
imagery	pearls and teeth; powerful spear; milky ocean
relationships	Murugan and Siva; Murugan and Arjuna
keywords	Murugan, devotion, protection, spirituality

Table 6: Example annotated record from the Thiruppugazh dataset

Findings in Tamil Dialect Speech Recognition and Classification

B. Bharathi[1], **Bharathi Raja Chakravarthi**[2],
Shunmuga Priya Muthusamy Chinnan[2], **S. Saranya**[3], **S. Suhasini**[4]
[1]Sri Sivasubramaniya Nadar College of Engineering, Tamil Nadu, India
[2]Unit for Inclusive AI, Data Science Institute, University of Galway, Ireland
[3] St. Joseph's Institute of Technology, Tamil Nadu, India
[4]Saveetha Engineering College, Tamil Nadu India
bharathib@ssn.edu.in, bharathi.raja@universityofgalway.ie

Abstract

As part of DravidianLangTech-2026, we provide a overview of Shared Task on Dialect-based Speech Recognition and Classification in Tamil. Creating reliable system for Tamil dialect identification from audio signals and dialect-aware Automatic Speech Recognition (ASR) is the main goal of the joint work. Dialect-based Tamil Speech Recognition and Tamil Dialect Classification from Speech are the two subtasks that make up the task. 5,134 audio recordings in four Tamil dialects: Southern, Northern, Western, and Central-spanning 9 hours and 22 minutes make up the training dataset. There are 579 audio samples in the test set, totaling almost two hours in length. The shared task involved 17 teams in total. For speech recognition and dialect classification, the top-performing system obtained a Word Error Rate (WER) of 0.51 and a macro F1-score of 0.79, respectively. The findings emphasize the difficulties in understanding Tamil speech due to dialectal diversity and set solid foundations for further study on low-resource dialect-aware ASR systems.

1 Introduction

Dialect-aware Tamil Speech Recognition and Dialect Classification, a shared task initially established to evaluate intra-language dialect modeling in a low-resource environment, is presented in this work. In order to close a significant gap in dialect-aware speech processing, the work attempts to methodically assess the robustness of automated speech recognition (ASR) systems and dialect identification approaches across four important Tamil dialects.

Over 75 million people worldwide speak Tamil, a traditional Dravidian language that makes an interesting case study for dialect-aware speech processing. Despite having a standardized literary form, Tamil's spoken dialects which include Southern, Northern, Western, and Central dialect groups differ significantly depending on the locale. These varieties of speech differ in lexical substitution patterns, prosodic contours, vowel length realization, and consonant articulation. In both ASR and dialect identification tasks, this variance raises inter-dialect confusability and generates auditory ambiguity.

Tamil is characterized by a significant amount of phonetic, lexical and prosodic variation between the geographical areas. These differences have a considerable impact on the speech recognition systems and dialect identification systems (Nanmalar et al., 2019). Although there are standard Tamil ASR systems, there are few dialect-sensitive benchmarks and evaluation platforms.

The past decades have seen considerable improvement in the performance of ASR in different languages due to the development of deep neural networks and self-supervised learning of speech representation (Radford et al., 2023). The advancements have made it possible to have the large- scale multilingual models that can cope with the different acoustical conditions (Ardila et al., 2020). Most current systems however just consider languages as acoustically homogenous objects, ignoring systematic intra-language variation like regional dialects. That the dialectal variation can significantly impair the ASR accuracy, particularly when the training data is imbalanced or has no phonetic variation in the region.

The development of deep learning and massive pre-trained speech models has allowed the advancement of Automatic Speech Recognition (ASR) systems (Baevski et al., 2020). Nonetheless, dialectal diversity has been one of the greatest challenges especially to the morphologically rich and low-resource languages like Tamil language(Nanmalar et al., 2022).

Although the core ASR activities have been extensively studied in Indian languages including Tamil, dialect level benchmarking is a mostly

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 71–79
July 3, 2026 ©2026 Association for Computational Linguistics

under-researched area (Srivastava et al., 2018). Available datasets normally address standard or broadcast speech and are not usually dialect annotated. Moreover, speech-based dialect identification in a monolingual environment is also an unsolved task.

This shared task solves these shortcomings by providing a curated benchmark data set of four Tamil dialects and assessing two complementary tasks: dialect-aware speech recognition and dialect classification. The task will contribute to improving the research in dialect-robust ASR and intra-language speech modeling by offering standardized metrics of evaluation and the competitive leaderboard environment. The task's objectives are:

- Promote dialect-aware Tamil ASR research
- Present a standardized dataset for assessment.
- Develop both baseline and cutting-edge performance standards.
- Encourage the study of dialect modeling with limited resources.

2 Related Work

Automatic Speech Recognition (ASR) for Tamil has evolved from the Hidden Markov Model (HMM)-Gaussian Mixture Model (GMM) frameworks to current end-to-end neural networks. Early systems used acoustic characteristics which were handcrafted like the MFCCs with statistical language models. The advent of deep learning saw the development of recurrent neural networks and attention-based encoder-decoder models that achieved much higher levels of recognition accuracy especially on morphologically rich languages.

Latest developments in self-supervised learning have also changed the speech processing. Representation learning using raw audio is made possible through such models as Wav2Vec 2.0 (Baevski et al., 2020; Hsu et al., 2021; Chen et al., 2022), which use large-scale unlabeled audio data to train. The pretrained models have been particularly useful in the low resource languages such as Tamil as they do not rely on large labeled corpora. Most more recently, Whisper, a large multilingual encoder decoder model, had been trained with weak supervision, showing good zero-shot and fine-tuning results on a wide range of languages, and among different dialects within identical languages.

Identifying dialect in the speech is even more difficult than regular language identification because the phonetic, prosodic and lexical differences in the same language are sometimes very slight. Previously used methods were based on statistical classifiers and manual crafted acoustic features, but newer techniques use deep neural networks and speaker embedding models as ECAPA-TDNN do (Desplanques et al., 2020). Both text-based and acoustic-based approaches have been examined and transformer-based encoders are always more effective than shallow approaches in the ability to capture dialect-specific features.

Dialect-rich languages Low-resource speech modeling is a developing field of study, especially when modeling low-resource languages. Multilingual pretraining and transfer learning have demonstrated a high possibility of enhancing generalization between dialects through the use of common acoustic representations. More computationally efficient fine-tuning models like LoRA(Hu et al., 2022) also impose lower demands on computational resources and achieve good performance, which is why they are also considered in dialect-aware adaptation.

According to (Srivastava et al., 2018), the development of ASR systems in Indian languages in limited data conditions is difficult. Likewise, (Ismail, 2020) surveyed language and dialect identification systems by focusing on the problem of acoustic similarity and the issue of class imbalance.

Studies that have been conducted on multilingual and transfer learning strategies have revealed major advancements in the low-resource context. (Yadav and Sitaram, 2022) conducted a review of multilingual ASR models and have shown that self-supervised pretrained models are effective. (Madhavaraj and Ramakrishnan, 2019) also demonstrated that data pooling and multi-task learning are more effective when it comes to the performance with a variety of low-resource languages.

The creation of the powerful ASR systems has been enabled by large multilingual corpora like Common Voice (Ardila et al., 2020). In the case of Tamil in particular, a model of ASR, end-to-end, was proposed by (R et al., 2023; Changrampadi et al., 2022) suggested a multilingual Indian speech corpus, which can be used in future studies.

By combining speech recognition with higher-level semantic processing, these approaches highlight the importance of robust acoustic modeling for dialectal speech in real-world scenarios. How-

ever, existing real-time systems primarily focus on end-to-end application development rather than systematic benchmarking across multiple dialects (Saranya et al., 2025).

Although recent methods of self-supervised and multilingual ASR models have been developed, the speech processing in Tamil dialect is not well explored. The current systems are mostly based on standard Tamil and scanty consideration on systematic benchmarking among dialects has been done. The dialect classification and ASR interaction are not thoroughly considered in unified conditions and publicly available dialect-specific datasets are still rare.

To reduce such gaps, the given shared task offers a common standard of dialect-conscious ASR and speech-based dialect recognition in Tamil that would allow systematic comparative analysis of the modeling approaches under equalized assessment conditions.

3 Task Description

The two complementing subtasks of the shared task, which focuses on dialect-aware speech processing for Tamil, are intended to assess transcription robustness and dialect discrimination capability. In a low-resource environment, the activities are designed to promote the creation of models that can manage intra-language variance. The speech-based dialect classification is significantly more difficult than a Text-based dialect identification, because the models have to account for the subtle variations in the acoustic, phonetic and prosodic aspects directly from the speech signal. Speech-level dialect recognition presents another challenge besides transcribed text, as it involves the recognition of variations in speaker, pronunciation, and recording context.

Input: Participants receive the following: 16 kHz sampling rate, variable-duration utterances, raw speech audio files in WAV format, and dialect labels that are exclusive to the training data.

3.1 Subtask 1: Dialect-Aware Tamil Speech Classification

Automatic dialect classification from speech signals is the main objective of Subtask 1. Systems are required to categorize a Tamil audio recording into one of four predetermined dialect groups. The dataset includes four major dialect groups:

- Southern_Dialect
- Northern_Dialect
- Western _Dialect
- Central_Dialect

Classification of these dialects is a challenging auditory discriminating challenge due to their systematic phonetic and prosodic variance.

Evaluation Metrics Systems are evaluated using:

- Macro F1-score is the thing we look at

We picked Macro F1 because it is fair to all dialect classes. This is especially important when some classes have a lot things in them than others. Macro F1 makes sure that each dialect class is treated the same so the classes with a lot of things in them do not get all the attention. This way Evaluation Metrics like Macro F1 are really good, for systems and dialect classes.

3.2 Subtask 2: Dialect-Aware Tamil Speech Recognition

Subtask 2 is to predict the corresponding transcription from a Tamil audio recording. In contrast to traditional ASR benchmarks, which presume standardized speech, this task specifically assesses recognition ability across a variety of dialects.

The test data does not include transcriptions.

Evaluation Metrics Performance is evaluated using: Word Error Rate (WER) as the primary metric WER = (Substitutions + Insertions + Deletions) / Total Reference Words

Word Error Rate is the number of mistakes, which are substitutions, which is when we use the wrong word, insertions, which is when we add an extra word, deletions, which is when we leave out a word. The purpose of the two subtasks is to compliment each other: The robustness of transcription under dialect variation is assessed in Subtask 1. The explicit modeling of dialect identity is assessed in Subtask 2.

Figure 1 presents the classification of the dialects and sample sentences considered in this task. They offer a thorough evaluation of dialect-aware speech modeling when combined. Although dialect cues may be implicitly learned by ASR systems, discriminative modeling of fine-grained audio variations is necessary for dialect classification. Deeper understanding of dialect confusability and representation learning is made possible by joint analysis of the two tasks.

S.No.	Name of the Dialect	Region	Example sentence
1	Northern_Dialect	Madras, Chengalpet and North Arcot Districts	சாப்டுறதுக்கு எதனா வாங்கின்னு வா. (Saapdrathukku edhana vaanginnu vaa.)
2	Southern_Dialect	Madurai, Ramnad, Tirunelveli and Kanyakumari	ஏல எப்பல வந்த? (yela eppala vantha)
3	Western_Dialect	Salem, Dharmapuri, Coimbatore and Nilgiris Districts	இன்னைக்கு என்னங்க ஒரே உப்பசமா இருக்குதுங்க. (Innaikku ennanga ore uppasama irukkudunga.)
4	Central_Dialect	Tiruchirappalli, Thanjavur and South Arcot Districts	ரெண்டு பேரும் அடிச்சுக்காதீங்க செத்த செவனேனு இருங்க (Rendu perum adichukkadheenga seththa sevanenu irunga.)

Figure 1: Dialect classifications with sample sentences

4 Dataset Description

Table 1: Training Data Statistics

Dialect	Audio_Files	Duration
Southern_Dialect	1427	2:44:30
Northern_Dialect	1696	3:29:15
Western_Dialect	1126	1:59:59
Central_Dialect	885	1:08:18
Total	5134	9:22:02

The training dataset consists of 5,134 audio recordings spanning 9 hours and 22 minutes across four Tamil dialects[(Bharathi et al., 2025).

Table 2: Test Data Statistics

Dialect	Audio_Files	Duration
Test Data	579	02:04:05

All audio files were sampled at 16 kHz and provided in WAV format. Table 1 and 2 summarizes the dataset distribution across dialects.

5 Participation

In the Dialect based Tamil speech recognition and classification shared task, 17 teams from academic institutions participated. The increasing interest in dialect-aware speech processing and low-resource ASR benchmarking is reflected in the participants' variety. Teams that took part used a variety of modeling techniques, including lightweight classification frameworks, hybrid acoustic models, and pretrained transformer-based architectures.

The competition followed a common evaluation procedure, with training set and test set of 579 audio examples (about 2 hours of speech). The final system submissions were tested on the test set without lable to avoid any kind of overfitting.

5.1 Modeling Approaches

Participants explored several modeling paradigms:

The majority of the best-performing systems utilized large pretrained models like Whisper encoder-decoder models or wav2vec/XLSR fine-tuning pipelines. These models had the following advantages:

- Self-supervised pretraining on large multilingual datasets
- Strong contextualized speech representations
- Fine-tuning on dialect-specific Tamil datasets

Transformer models had better generalization capabilities, especially in Subtask 1 (ASR).

Team CHMOD_777 fine-tuned Wav2Vec 2.0 Base for dialect classification directly from speech.

For ASR, they fine-tuned Whisper Small. Their approach relies on supervised adaptation of pretrained multilingual speech encoders.

Team TamilVoiceLab used Whisper Base for ASR and extracted embeddings from Wav2Vec 2.0 for dialect classification, followed by a softmax classification layer. Their system focuses on transfer learning under limited dialectal data.

Team Dialectmind experimented with Whisper Small and Whisper Medium models for ASR. For dialect identification, they trained a text classifier on ASR-generated transcripts using transformer-based encoders.

Team CUET_InferX fine-tuned Whisper Small for transcription and used MuRIL (Multilingual BERT for Indian Languages) for dialect classification from text outputs. Their pipeline separates acoustic and linguistic modeling.

Team Violet_Vortex used Whisper Base for ASR and applied an LSTM-based classifier over acoustic embeddings for dialect classification. Their approach evaluates sequential neural models for dialect discrimination.

Team AITamilDialect adopted a multi-task learning framework built on Wav2Vec 2.0, jointly optimizing ASR and dialect classification objectives with task-specific output heads. **Team HAG Signals** fine-tuned Whisper Small for ASR and experimented with beam search decoding strategies. For dialect classification, they used a fully connected classifier on speech embeddings.

Team DLRG employed parameter-efficient fine-tuning (LoRA) on Whisper Base for ASR. For dialect classification, they used Wav2Vec 2.0 embeddings with a linear classifier.

Team IITK_SpeechScape used Conformer-based ASR architecture initialized with pretrained acoustic representations. For dialect classification, they fine-tuned WavLM.

Team Sync2 focused exclusively on dialect classification using ECAPA-TDNN speaker embedding architecture, adapted to capture dialectal acoustic variations.

Team CODERS implemented a unified framework using Wav2Vec 2.0 as a shared encoder backbone, with separate heads for ASR and dialect classification.

Team Wave2Word used Whisper Small for ASR and a transformer-based classifier for dialect prediction from transcriptions. In their second submission, Wave2Word experimented with WavLM for dialect classification and compared performance against their Whisper-based pipeline.

Team Wise adopted a lightweight dialect classification model built on Wav2Vec 2.0 Base, followed by a fully connected neural classifier.

Team SpokenRoots used HuBERT Base for acoustic feature extraction and trained a supervised classifier for dialect identification.

Team GigitAI employed an ECAPA-TDNN embedding model, originally designed for speaker recognition, adapted for dialect classification tasks.

Team Azrael developed a computationally efficient dialect classifier based on Wav2Vec 2.0 Small, optimized for low-resource deployment environments.

The 17 participating teams' architectures and training methods are compiled in Table 3. We note that the most popular backbone models were Whisper and Wav2Vec 2.0, whereas only a small number of teams investigated multi-task learning and parameter-efficient fine-tuning methods.

6 Results Overview

The official results of the evaluation of both subtasks of the shared task are presented (i) Dialect Classification and (ii) Dialect-aware Automatic Speech Recognition (ASR). There were 17 teams in the competition that completed at least one subtask and handed systems that were rated on a held-out test set. The evaluation of Subtask 1 was conducted with Macro F1-score to consider the imbalance in classes among the dialects, The evaluation for Subtask 1 was based on macro F1-score since there was no uniform distribution of the dialect classes. Unlike Micro-F1, Macro F1-score measures the performance separately for each class and then averages the results, thus assigning equal weights to all dialect categories irrespective of their size, whereas in Subtask 2 the evaluation of the performance was by Word Error Rate (WER) where smaller numbers showed better performance.

In Subtask 1, Wise scored the highest Macro F1-score of 0.79, a score that is far ahead of the one that second-ranked system scored. The performance distribution indicates that there is a significant difference between the highest performance system and the rest of the submissions, which implies a difference in the ability to model dialectal acoustic variation.

Table 3: Overview of modeling approaches used by the 17 participating teams.

Team	Whisper Small	Whisper Base	Whisper Medium	Wav2Vec 2.0	WavLM	HuBERT	ECAPA-TDNN	MuRIL	LoRA	Multi-task	Data Aug.
CHMOD_777(Karunanidhi and Arumugam, 2026)	✓			✓							✓
TamilVoiceLab(Priya and Bharathi, 2026)		✓		✓							
Dialectmind(Gayathri and Bharathi, 2026)	✓		✓					✓			
CUET_InferX(Semon et al., 2026)	✓							✓			
Violet_Vortex		✓		✓							✓
AITamilDialect(Varalakshmi and Bharathi, 2026)				✓						✓	
HAG_Signals	✓			✓							✓
DLRG_1(Abhinav et al., 2026)		✓		✓					✓		
IITK_SpeechScape(Sekar et al., 2026)					✓						
Sync2							✓				
CODERS				✓						✓	
Wave2Word_1 (Naswan and Ahmad, 2026)	✓										
Wave2Word_2(Naswan and Ahmad, 2026)					✓						
Wise(Ganesh Sundhar et al., 2026)				✓							
SpokenRoots						✓					
GigitAI							✓				
Azrael(J et al., 2026)				✓							

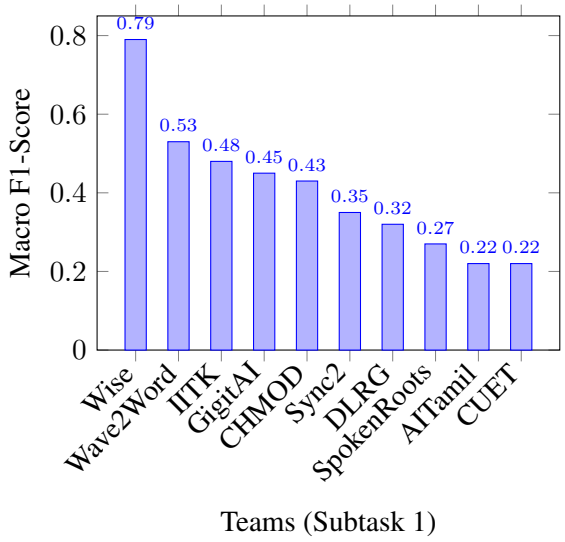


Figure 2: Performance trend of top teams in Subtask 1 (Dialect Classification).

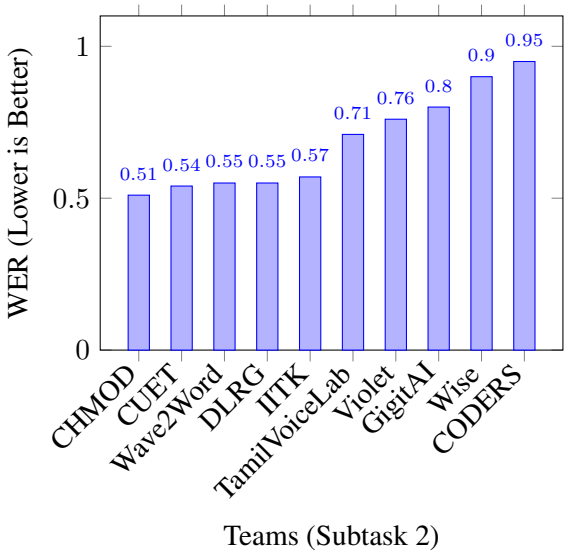


Figure 3: Performance distribution in Subtask 2 (Dialect-aware ASR).

In the case of Subtask 2, CHMOD_777 had the lowest WER of 0.51 with CUET_InferX (0.54) and Wave2Word (0.55) being close by. The most successful systems were mainly based on highly tuned systems using transformer-based ASR models, specifically Whisper models. Nevertheless, some submissions obtained a WER value exceeding 0.90, which means that there are still significant difficulties with speech recognition with dialect awareness in low-resource.

In Subtask 1, Wise obtained the highest Macro F1-score, as indicated in Table 4.

The official WER ranking for Subtask 2 is shown in Table 5.

Figures 2 and 3 present the performance trends for both subtasks. The study of the model architectures shows that there is a definite correlation between the pretrained transformer-based speech models and the increased performance. Whisper variants based systems were always dominant in Subtask 2 and CHMOD_777 was the lowest WER

Table 4: Official ranking of Subtask 1 (Dialect Classification) on the test set.

Rank	Team Name	Run	Precision	Recall	F1_Score
1	Wise	Run2	0.89	0.76	**0.79**
2	Wave2Word	Run1	0.54	0.54	0.53
3	IITK_SpeechScape	Run2	0.54	0.50	0.48
4	GigitAI	Run1	0.46	0.49	0.45
5	CHMOD_777	Run1	0.41	0.47	0.43
6	Sync2	Run3	0.40	0.39	0.35
7	DLRG	Run1	0.33	0.37	0.32
8	SpokenRoots	Run2	0.30	0.28	0.27
9	AITamilDialect	–	0.23	0.22	0.22
9	CUET_InferX	Run2	0.23	0.30	0.22
10	CODERS	Run1	0.21	0.22	0.18
11	Azrael	Run1	0.08	0.25	0.12

Table 5: Official ranking of Subtask 2 (Dialect-aware ASR) based on Word Error Rate (WER) on the test set

Rank	Team Name	WER
1	CHMOD_777_Run1	0.51
2	CUET_InferX_Run1	0.54
3	Wave2Word_Run1	0.55
3	DLRG_Run1	0.55
4	IITK_SpeechScape_Run1	0.57
5	TamilVoiceLab_Run1	0.71
6	VIOLET_VORTEX_Run1	0.76
7	GigitAI_Run3	0.80
8	Wise_Run2	0.90
9	CODERS_Run1	0.95
10	AITamilDialect	1.00
10	HAG_Signals	1.00
10	DialectMind_Run1	1.00

(0.51). Likewise, encoders based on transformers (including Wav2Vec 2.0 and WavLM) were also able to achieve competitive performances in dialect classification.

However, this trend was not observed in systems based only on lightweight or shallow classifiers, and in this case, Macro F1-scores were significantly lower, indicating that dialect discrimination is indeed greatly improved with the help of deep contextual acoustic representations. Surprisingly, the teams which implemented common encoder backbone on both subtasks (e.g. Wave2Word and DLRG) exhibited equal classification and ASR performance, which suggests that a strong acoustic modeling can generalize to similar speech tasks.

Interestingly, the best performing system in Subtask 1 did not appear to perform competitively in Sub task 2 and vice versa. It indicates that the modeling attributes of dialect classification and dialect-aware ASR are different. Discriminative acoustic representations with good capturing of dialect difference is the major advantage of dialect classification, while accurate lexical and phonetic transcription ability is the key to ASR performance. As a result, models learnt from one task may not necessarily perform well on the other task.

In general, the findings re-establish the roles of the large pretrained multilingual speech models in low-resource dialectal settings.

7 Limitations

Although, this common task can be regarded as a good source of insight into dialect-conscious speech recognition and classification on the Tamil language, there are still a number of limitations.

To begin with, the size of the dataset is quite small in comparison with the large-scale benchmarks of ASR. Though several dialects were involved, the quantity of speech data of each dialect might not be large enough to comprehensively reflect the intra-dialect variation, the speaker heterogeneity, and the environmental noise factors.

Second, the task was performed on one held-out test set, which might not necessarily be applicable in the real world of deployment. The spontaneous conversation speech can have dialect variations that may introduce new challenges outside of the edited dataset.

Third, the majority of systems involved in it were heavily dependent on the pretrained multilingual transformer models, including Whisper and Wav2Vec 2.0. These models showed good performance but their computational needs can confine its use in low resource or on-device deployment environments.

Lastly, this combined task concentrated solely

on acoustic as well as transcription-based modelling. Future research may look into multimodal cues, language modeling that is dialect-aware and fairness-aware across demographic boundaries.

Although these are the constraints, the activity forms a brilliant foundation to further studies of dialect finely-sensitive speech technologies in Tamil.

8 Conclusion

This shared task on Dialect-based Speech Recognition and Classification in Tamil highlights both the progress and remaining challenges in dialect-aware speech technologies. In 17 competing teams, transformer-based pretrained models, and specifically Whisper and Wav2Vec 2.0, were the most widespread architectures and they showed impressive generalization in low-resource dialectal settings. Although there is a top-of-class competitive performance, there is a significant difference between the highest-ranked systems and the lowest-ranked submissions, especially in the territory of ASR, wherein various models appeared incapable of achieving WER to less than 0.70. This means that the dialectal variation remains a major challenge to the acoustic modeling and adaptation.

The research directions of the future are the investigation of multi-task learning models, which will optimize transcription and dialect classification, parameter-efficient fine-tuning strategies, and dialect-aware decoding models. The assignment highlights the need to further expand datasets and enhance the representation learning to enhance inclusivity in the underrepresented dialects speech technologies.

Acknowledgments

The authors, Bharathi Raja Chakravarthi was funded by a research grant from Research Ireland under grant number SFI/12/RC/2289_P2 (Insight_2).

References

Gulisetty Abhinav, Tanisha Nanda, Ramesh Kannan R, and Ratnavel Rajalakshmi. 2026. Dlrg@dravidianlangtech 2026: Dual-purpose whisper adaptation for tamil dialect identification and dialectal speech recognition. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:12449–12460.

B Bharathi, S Saranya, P Vijayalakshmi, and T Nagarajan. 2025. Multi-dialect speech corpus creation for enhancing tamil automatic speech recognition. *Circuits, Systems, and Signal Processing*, pages 1–19.

Mohamed Hashim Changrampadi, A. Shahina, M. Badri Narayanan, and A. Nayeemulla Khan. 2022. End-to-end speech recognition of Tamil language. *Intelligent Automation & Soft Computing*, 32(2):1309–1323.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Yao Qian, and Furu Wei. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. In *Proceedings of Interspeech*, pages 1503–1507.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. 2020. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. In *Proceedings of Interspeech*, pages 3830–3834.

S Ganesh Sundhar, N Hari Krishnan, G Gnanasabesan, KP Suriya, and G Jyothish Lal. 2026. Wise@dravidianlangtech 2026: Dialect-aware tamil speech classification and recognition via cross-pipeline embedding transfer. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

K Gayathri and B Bharathi. 2026. Dialect-mind@dravidianlangtech 2026: Zero-shot dialectal tamil automatic speech recognition using a large pre-trained conformer model. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*.

Tanvira Ismail. 2020. A survey of language and dialect identification systems. *Adalya Journal*, 9(1).

Janish Andrin J, Mohammed Sahil S, and S Saranya. 2026. Azrael@dravidianlangtech 2026: Dialect-sensitive automatic speech recognition and classification for tamil. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Arunaggiri Pandian Karunanidhi and Prabalakshmi Arumugam. 2026. Chmod_777@dravidianlangtech 2026: Tamil-adapted whisper and mms for dialect speech recognition and classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

A Madhavaraj and A G Ramakrishnan. 2019. Data-pooling and multi-task learning for enhanced performance of speech recognition systems in multiple low resourced languages. In *2019 National Conference on Communications (NCC)*, pages 1–5.

M. Nanmalar, P. Vijayalakshmi, and T. Nagarajan. 2019. Literary and colloquial dialect identification for Tamil using acoustic features. In *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, pages 1303–1306.

M. Nanmalar, P. Vijayalakshmi, and T. Nagarajan. 2022. Literary and colloquial tamil dialect identification. *Circuits Syst. Signal Process.*, 41(7):4004–4027.

Ruwad Naswan and Shadab Tanjeed Ahmad. 2026. Wave2word@dravidianlangtech 2026: Whistam: A unified framework for dialect based tamil speech recognition and classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

S. B. Priya and B. Bharathi. 2026. Tamilvoice-lab@dravidianlangtech 2026: Investigating whisper tamil large-v2 for dialectal tamil speech recognition. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Nithya R, Malavika S, Jordan F, Arjun Gangwar, Metilda N J, S Umesh, Rithik Sarab, Akhilesh Kumar Dubey, Govind Divakaran, Samudra Vijaya K, and Suryakanth V Gangashetty. 2023. Spring-inx: A multilingual indian language speech corpus by spring lab, iit madras. *Preprint*, arXiv:2310.14654.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. *Proceedings of the 40th International Conference on Machine Learning (ICML)*.

S Saranya, B Bharathi, S Gomathy Dhanya, and Aishwarya Krishnakumar. 2025. Real-time continuous tamil dialect speech recognition and summarization. *Circuits, Systems, and Signal Processing*, 44(4):2855–2881.

G Srishtik Sekar, Harissh Ragav Dhamodaran, Kishore Shankar S, Balasubramanian Palani, and R. Tharaniya Sairaj. 2026. Iiitk_speechscape@dravidianlangtech 2026: Dialect based speech recognition and classification using speech foundation models and deep learning techniques. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Md. Ashraful Islam Semon, Jihadul Islam, Ratnajit Dhar, and Hasan Murad. 2026. Cuet_inferx@dravidianlangtech 2026: Shared task on dialect based speech recognition and classification in tamil. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Brij Mohan Lal Srivastava, Sunayana Sitaram, Rupesh Kumar Mehta, Krishna Doss Mohan, Pallavi Matani, Sandeepkumar Satpal, Kalika Bali, Radhakrishnan Srikanth, and Niranjan Nayak. 2018. Interspeech 2018 Low Resource Automatic Speech Recognition Challenge for Indian Languages. In *Proc. 6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2018)*, pages 11–14.

K Varalakshmi and B Bharathi. 2026. Aitamil-dialect@dravidianlangtech 2026: Zero-shot whisper and wav2vec2 embedding-based tamil speech recognition and dialect classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Hemant Yadav and Sunayana Sitaram. 2022. A survey of multilingual models for automatic speech recognition. *Preprint*, arXiv:2202.12576.

Findings of the Shared Task on Hope Speech Detection in Tulu

Durairaj Thenmozhi[1], Anusha M D[2], Raksha Adyanthaya[2], Rathnakara Shetty P[2], Parameshwar R. Hegde[2], Mohammed Fadhel Aljunid[3], Prasanna Kumar Kumaresan[4], Bharathi Raja Chakravarthi[4]

[1]Sri Sivasubramaniya Nadar College of Engineering, Tamil Nadu, India
[2]YIASCM, Yenepoya (Deemed to be University), Mangalore, India
[3]Moodlakatte Institute of Technology, Karnataka, India
[4]Unit for Inclusive AI, Data Science Institute, University of Galway, Ireland

Abstract

Hope Speech Identification is the process of detecting positive, supportive, and encouraging language in text. It focuses on identifying content that promotes unity, inclusiveness, and resilience. Identification of hope speech helps supports mental well being, create healthier online environments, counter hate speech, and promote positive digital communication. This shared task hope speech detection in code-mixed Tulu language as part of DravidianLangTech @ ACL 2026, focuses on both the coarse-grained hope tone classification and the fine-grained hope type classification tasks. There are 11 teams participated in the tasks and submitted several runs for both the tasks. The teams are ranked based on the macro-F1 score.

1 Introduction

The proliferation of social media platforms has significantly transformed digital communication, enabling individuals to express emotions, opinions, and social support at an unprecedented scale (Chakravarthi et al., 2021). While substantial research in Natural Language Processing (NLP) has focused on detecting harmful content such as hate speech, offensive language, and misinformation, comparatively limited attention has been given to identifying positive and supportive expressions, particularly hope speech (Bali et al., 2014). Hope speech plays a vital role in promoting emotional well-being, strengthening social cohesion, and fostering constructive discourse in online communities (Conneau et al., 2020).

Automatic detection of hope-related expressions presents unique challenges, especially in multilingual and code-mixed environments (Kakwani et al., 2020). Most existing studies on hope speech detection have concentrated on high-resource languages such as English, Hindi, and Tamil (Chakravarthi et al., 2020). However, low-resource Dravidian languages remain underexplored due to the scarcity of annotated datasets and benchmark tasks (Chakravarthi et al., 2021). Tulu, a Dravidian language spoken predominantly in coastal Karnataka, represents one such underrepresented language, particularly in code-mixed contexts involving Tulu, English, and Kannada. The informal and transliterated nature of social media text further complicates computational modeling.

To address this research gap, we introduce the Hope Speech Detection shared task at DravidianLangTech@ACL 2026. The shared task presents the first benchmark dataset for hope speech detection in code-mixed Tulu and proposes a multi-level classification framework comprising two tracks: (i) coarse-grained hope tone classification and (ii) fine-grained hope type classification. The coarse-grained task focuses on identifying the overall emotional orientation of user comments with respect to hope, while the fine-grained task captures nuanced distinctions such as optimistic hope, realistic hope, inspiring hope, fading hope, and hopelessness.

The objective of this shared task is threefold:

1. To encourage research in low-resource Dravidian languages,

2. To benchmark computational approaches for emotionally nuanced language understanding, and

3. To foster the development of robust models capable of handling code-mixed and socially contextualized text.

The task attracted multiple participating teams employing diverse methodologies ranging from traditional machine learning models to transformer-based architectures and ensemble approaches. This

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 80–87
July 3, 2026 ©2026 Association for Computational Linguistics

paper presents an overview of the dataset, task formulation, participating systems, evaluation results, and key findings from the shared task.

2 Related Work

Hope speech detection, aimed at identifying uplifting, inspiring, and encouraging phrases on the Internet, has become an important problem in Natural Language Processing. Initial studies, as proposed by Chakravarthi, introduced the HopeEDI dataset, a multilingual collection of YouTube comments in English, Tamil, and Malayalam, and provided baseline results with the help of traditional machine learning models, including Support Vector Machines (SVM), Logistic Regression, Decision Trees, and Naive Bayes using the TF-IDF features. The study of (Divakaran et al., 2024) investigated the use of machine learning and transformer-based models, such as SVM, Logistic Regression, BERT, and DistilSpanBERT, for the detection of hope speech in Spanish and English as part of the shared task IberLEF 2024. The transformer models outperformed conventional machine learning techniques, achieving macro F1-scores of up to 0.82 for binary classification and 0.64 for multiclass classification. This improved Hope speech detection in multilingual environments through recent shared task initiatives. Subsequent work by Chakravarthi and Muralidaran explored deep learning-based approaches like Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), and CNN-LSTM approaches, which achieved weighted F1-scores of 0.93 in English, 0.84 in Malayalam, and 0.56 in Tamil, better than the traditional machine learning baselines, and found that contextual modeling through deeper neural networks was important to hope speech recognition. Despite these advancements, hope speech detection remains underexplored for low-resource languages like tulu. To address this gap, the present shared task introduces a new dataset and a variety of evaluation frameworks for hope speech detection.

3 Task Description

Shared task on hope speech detection in code-mixed Tulu language consists of two subtasks namely coarse-grained hope tone classification and fine-grained hope type classification.

- Task 1: Coarse-Grained Hope Tone Classification

  The task involves classifying social media comments' emotional tone by examining hope-related expressions. Comments must be categorized based on presence, absence, or ambiguity of hope.

  1. Encouraging: Text that conveys a hopeful, positive, or emotionally supportive tone. This may include affirmations, motivational statements, reassurance, or empathy.
  2. Discouraging: Content that reflects hopelessness, negativity, emotional withdrawal, despair, or defeat.
  3. Uninvolved: Neutral or informational content that does not express emotional engagement related to hope.
  4. Blended Tone: Text containing both hopeful and discouraging elements, resulting in emotional ambiguity or complexity.

  This coarse-grained classification enables systems to capture the general emotional orientation of user-generated content.

- Task 2: Fine-Grained Hope Type Classification

  The second task involves a more detailed classification of hope-related expressions. This fine-grained categorization is applied when hope signals are present (e.g., in Encouraging or Blended Tone categories, and certain Discouraging cases reflecting diminishing hope).

  The following labels are defined:

  1. Optimistic Hope: A confident belief that positive outcomes will occur in the future.
  2. Realistic Hope: Hope grounded in practical expectations and achievable outcomes.
  3. Inspiring Hope: Motivational expressions intended to encourage others toward a better future.
  4. Fading Hope: Expressions indicating that hope is gradually diminishing due to persistent challenges.
  5. Hopelessness: A strong sense of despair characterized by the belief that no positive outcome is possible.

This fine-grained task allows models to move beyond binary sentiment detection and capture subtle emotional distinctions within hope-related discourse.

Participants were given training, development, and test datasets to help them create classification models. The goal of this objective was to promote research in under-resourced languages and investigate novel techniques to offense detection in multilingual and code-mixed texts. Table 1 and Table 2 present illustrative examples from the dataset for Task 1 and Task 2 respectively, along with their English translations to aid understanding of the code-mixed Tulu text and the distinctions between label categories.

4 Data Description

The dataset is organized into two tasks: Task 1 (Coarse-Grained Hope Tone Classification) and Task 2 (Fine-Grained Hope Type Classification). The data for both sub tasks is divided into training, development, and test sets while maintaining representative label distributions.

Task 1 consists of 7,504 instances, comprising 8,559 comments and 9,972 sentences with a total of 69,798 words. Task 2 contains 4,550 instances, including 3,014 comments and 32,511 sentences with 671,420 total words.

Each sample was first annotated by two native Tulu speakers who also have a background in NLP research, giving them both the linguistic intuition and technical understanding needed for the task. The annotated samples were then passed on to linguistic experts from university language faculties, who bring years of teaching experience in the language. These experts reviewed the labels, clarified ambiguous cases, and ensured consistency across all categories. Inter-annotator agreement measured using Krippendorff's Alpha (Krippendorff, 2018) is 0.892 for Task 1 and 0.901 for Task 2, indicating strong and excellent annotation reliability respectively.

5 Participants Methodology

A total of 11 teams participated in Task 1, coarse-grained hope tone classification and 9 teams participated in Task 2, fine-grained hope type classification. They all used methodologies ranging from traditional classifiers with TF-IDF vectorization to transformer models for the above mentioned tasks. The run submissions of each team are explained below.

5.1 CUET_Synthetica

For Task 1, three pre-trained transformers xlm-roberta-base, google/muril-base-cased (MuRIL), and mbert were fine-tuned by this team (Zaman et al., 2026) for classifying text into four hope speech categories. The text was preprocessed by removing unwanted characters, normalizing whitespace, and tokenizing with a 128-token maximum length. Class-weighted loss was used to handle class imbalance. Models were trained using the AdamW optimizer (learning rate: 2e-5, weight decay: 0.01) for up to 8 epochs with early stopping (patience=2) based on the macro F1 score. The predictions from all three models were combined using an ensemble approach, where their logits were averaged element-wise. The final predictions were made using the argmax of the averaged logits. The ensemble of all three models achieved a development macro F1 score of 0.622 on the development set, improving upon individual models (XLM-R: 0.591, MuRIL: 0.591, mBERT: 0.607).

For Task 2, the model used xlm-roberta-base, a pre-trained transformer, for fine-grained classification of text data into five categories: optimistic hope, realistic hope, inspiring hope, fading hope, and hopelessness. The data was preprocessed by cleaning and tokenizing the text, with labels converted to lowercase and mapped to numerical values. The model was fine-tuned with class-weighted loss to address class imbalance, using the AdamW optimizer with a learning rate of 2e-5 and weight decay of 0.01. Training was done for up to 8 epochs with early stopping after 2 epochs of no improvement. The model's performance was evaluated using macro F1 score, and a learning rate scheduler was applied with a warm-up ratio of 0.1.The model achieved a macro F1 score of 0.4349 on the development set.

5.2 SSN_HopeNetters

Team SSN_HopeNetters (A et al., 2026) fine-tuned the multilingual transformer XLM-RoBERTa for Task 1. Input texts are tokenized using the XLM-R tokenizer with a fixed maximum sequence length, and the model is trained to predict one of four tone labels. To address class imbalance across tone categories, they incorporate a class-weighted cross-entropy loss, ensuring improved learning for minority classes such as discouraging and blended hope.

Table 1: Illustrative Examples for Task 1 : Coarse-Grained Hope Tone Classification

Text (Code-Mixed Tulu)	English Translation	Label
Rathish mendon.. thank you so much anna.. and comment maltina prativoriyagla solmelu	Rathish Mendon, thank you so much brother, and my greetings to everyone who commented	Encouraging Hope
Audio days marre madee sari ijja	Even the audio these days is not good at all	Discouraging Hope
Namma tulundadu comedy first eppodu	Our Tulu content always puts comedy first	Uninvolved
pudar daane change malthini natakada,..	It will change on its own, this is just drama	Blended Hope

Table 2: Illustrative Examples for Task 2: Fine-Grained Hope Type Classification

Text (Code-Mixed Tulu)	English Translation	Label
Navin padilerna gotanaga Portand serial mast soku ethundu, anchina scripts malpule, jai thulu nadu	If Navin acts in it, the Portland serial will be really great, please write such scripts, long live Tulu Nadu	Optimistic Hope
Very nice. Baari yedded malpuva. Superb. Good Luck	Very nice. You are doing really well. Superb. Good Luck	Inspiring Hope
undu artists.. situation toodhu dakkini, super skit	These artists have understood the situation well, super skit	Realistic Hope
Tuluta nadut kannada borchitund materla tulut paternda yedde ettund.	While walking in Tulu, Kannada is being mixed in; the original Tulu pattern is gradually being lost	Fading Hope
bupper pura paterver mare karma. inchi sav da film manpvar nikulu. thu	Completely wasted effort, what bad karma. Those who make such films should just stop	Hopelessness

Table 3: Dataset Split for Task 1 and Task 2

Task	Train	Dev	Test	Total
Task 1 (Coarse)	5,252	1,126	1,126	7,504
Task 2 (Fine)	3,185	682	683	4,550

Table 4: Task 1 Label Distribution

Label	Count
Encouraging Hope	2,708
Uninvolved	2,657
Blended Hope	1,146
Discouraging Hope	993
Total	**7,504**

Table 5: Task 2 Label Distribution

Label	Count
Inspiring Hope	1,613
Hopelessness	1,338
Realistic Hope	719
Optimistic Hope	543
Fading Hope	337
Total	**4,550**

The model is optimized using AdamW with a low learning rate and extended training epochs.

For Task 2, they performed fine-grained classification of hope expressions into five semantically nuanced categories, focusing on emotionally subtle and imbalanced labels. They employ XLM-RoBERTa as the base encoder to capture multilingual and code-mixed representations of Tulu text. Given the severe label imbalance in the dataset, they apply a weighted loss function that penalizes misclassification of low-frequency hope types such as fading and realistic hope. The model is fine-tuned with a reduced learning rate, longer warm-up, and increased training epochs to stabilize optimization. Predictions are generated using a softmax output layer.

Table 6: Comparison of submitted systems for Task 1: Coarse-Grained Hope Tone Classification.

Team	Preprocessing	Data Balancing	External Dataset	Approach	XLM-RoBERTa	MBert	IndicBert	Muril	DistilBert	Traditional	Prompting
CUET_Synthetica (Zaman et al., 2026)	✓	✓	✗	Ensemble of transformers	✓	✓	✗	✓	✗	✗	✗
SSN_HopeNetters (A et al., 2026)	✗	✓	✗	Transformers with data balancing	✓	✗	✗	✗	✗	✗	✗
Hope_Alchemists (Johnson et al., 2026)	✓	✓	✗	LinearSVC with TFIDF	✗	✗	✗	✗	✗	✓	✗
Team Oryu (Moni et al., 2026)	✗	✓	✗	Transformers with oversampling	✓	✗	✗	✗	✗	✗	✗
SJM_MINDS (Shashirekha et al., 2026)	✓	✓	✗	LinearSVC with TF-IDF	✗	✗	✗	✗	✗	✓	✗
Tensor_Thunders	✓	✗	✗	Fine tuning transformers	✗	✗	✗	✗	✓	✗	✗
CUETEuropa	✗	✓	✗	Fine tuning transformers	✗	✗	✓	✗	✗	✗	✗
cantnlp (Li and Wong, 2026)	✗	✗	✓	Fine tuning transformers with external dataset	✓	✗	✗	✗	✗	✗	✗
MUCS (Shashirekha and A, 2026)	✓	✗	✗	Traditional classifier with TF-IDF	✗	✗	✗	✗	✗	✓	✗
NamDang	✗	✗	✗	DSPy framework with few-shot prompting	✓	✗	✗	✗	✗	✗	✓
PrimeLine (V et al., 2026)	✗	✗	✗	Fine tuning transformers	✓	✗	✗	✗	✗	✗	✗

Table 7: Comparison of submitted systems for Task 2: Fine-Grained Hope Type Classification.

Team	Preprocessing	Data Balancing	External Dataset	Approach	XLM-RoBERTa	MBert	IndicBert	Muril	DistilBert	Traditional
CUET_Synthetica (Zaman et al., 2026)	✓	✓	✗	Fine tuning transformers	✓	✗	✗	✗	✗	✗
SSN_HopeNetters (A et al., 2026)	✗	✓	✗	Transformers with weighted loss	✓	✗	✗	✗	✗	✗
Hope_Alchemists (Johnson et al., 2026)	✗	✓	✗	Fin tuning transformer with oversampling	✓	✗	✗	✗	✗	✗
SJM_MINDS (Shashirekha et al., 2026)	✓	✓	✗	Logistic Regression with TF-IDF	✗	✗	✗	✗	✗	✓
Tensor_Thunders	✓	✗	✗	Fine tuning transformers	✗	✗	✗	✗	✓	✗
CUETEuropa	✗	✓	✗	Fine tuning transformers	✗	✗	✓	✗	✗	✗
cantnlp (Li and Wong, 2026)	✗	✗	✓	Fine tuning transformers with external dataset	✓	✗	✗	✗	✗	✗
MUCS (Shashirekha and A, 2026)	✓	✗	✗	Traditional classifier with TF-IDF	✗	✗	✗	✗	✗	✓
PrimeLine (V et al., 2026)	✗	✗	✗	Fine tuning transformers	✓	✗	✗	✗	✗	✗

5.3 Hope_Alchemists

For Task 1, this team (Johnson et al., 2026) employed the LinearSVC classifier using class_weight="balanced" with a TF-IDF (word 1–2 grams + character 3–5 grams) features to train the model. For Task 2, they fine-tuned XLM-RoBERTa base for the classification. To address label imbalance, they used oversampling to equalize class counts and trained the transformer with focal loss and early stopping.

5.4 Team Oryu

This team (Moni et al., 2026) participated only in Task 1 and they fine-tuned the pretrained XLM-RoBERTa model to leverage its strong multilingual contextual representations for the Tulu dataset. The raw text was first cleaned and tokenized using the XLM-R tokenizer with a maximum sequence length of 192 to preserve contextual cues. To address class imbalance, they applied training data oversampling and optimized the model using focal loss, which improves learning for hard and minority-class examples. The model was trained with the AdamW optimizer, linear learning rate scheduling with warmup.

5.5 SJM_MINDS

This team (Shashirekha et al., 2026) implemented a Linear Support Vector Machine (Linear SVC) classifier for Task 1. Prior to feature extraction, text preprocessing was performed, including lowercasing and removal of URLs, user mentions, hashtags, and punctuation. Text data was transformed using TF-IDF representation with up to 8000 features, considering unigram and bigram combinations and removing rare terms occurring in less than two documents. To address class imbalance, class weights were set to "balanced".

For Task 2, the team used a Logistic Regres-

Table 8: Hope Speech Identification - Task 1 Results

S.no	Team Name	Run	Accuracy	Macro Precision	Macro Recall	Weighted Precision	Macro F1-Score	Rank
1	CUET_Synthetica	Run 1	0.67	0.58	0.58	0.69	0.58	1
2	SSN_HopeNetters	Run 1	0.64	0.56	0.56	0.68	0.55	2
3	Hope_Alchemists	Run 2	0.64	0.55	0.55	0.66	0.55	2
4	Team Oryu	Run 1	0.60	0.55	0.54	0.68	0.53	3
5	SJM_MINDS	Run 2	0.62	0.52	0.51	0.61	0.51	4
6	Tensor_Thunders	Run 1	0.65	0.53	0.51	0.62	0.51	4
7	CUETEuropa	Run 2	0.58	0.54	0.52	0.67	0.51	4
8	cantnlp	Run 1	0.66	0.50	0.51	0.62	0.50	5
9	MUCS	Run 2	0.57	0.47	0.47	0.57	0.47	6
10	NamDang	Run 1	0.50	0.43	0.43	0.52	0.41	7
11	PrimeLine	Run 1	0.60	0.30	0.41	0.44	0.34	8

Table 9: Hope Speech Identification - Task 2 Results

S.no	Team Name	Run	Accuracy	Macro Precision	Macro Recall	Weighted Precision	Macro F1-Score	Rank
1	CUET_Synthetica	Run 3	0.50	0.42	0.42	0.49	0.42	1
2	Tensor_Thunders	Run 1	0.54	0.48	0.41	0.52	0.41	2
3	Hope_Alchemists	Run 1	0.49	0.41	0.40	0.49	0.40	3
4	SSN_HopeNetters	Run 1	0.45	0.40	0.43	0.49	0.38	4
5	SJM_MINDS	Run 1	0.56	0.64	0.39	0.59	0.37	5
6	CUETEuropa	Run 2	0.45	0.36	0.36	0.45	0.36	6
7	cantnlp	Run 1	0.54	0.35	0.36	0.45	0.33	7
8	MUCS	Run 2	0.38	0.29	0.29	0.37	0.29	8
9	PrimeLine	Run 1	0.24	0.19	0.19	0.24	0.19	9

sion classifier with TF-IDF feature representation. The dataset was split into 80% training and 20% validation sets using a fixed random seed (random_state=42) to ensure reproducibility. The Logistic Regression model was trained with a maximum iteration limit of 1000. After validation, the model was retrained on the full training dataset and used to generate predictions for the test set prediction.

5.6 Tensor_Thunders

This team used the DistilBERT model for both Task 1 and Task 2 to perform text classification on the given dataset. The data was preprocessed and then the pre-trained DistilBERT model was fine-tuned by using the optimized hyperparameters.

5.7 CUETEuropa

This team used IndicBERT as the backbone architecture for both Task 1 and Task 2. The hope speech labels were mapped to numerical IDs using a bidirectional label encoding scheme. Text inputs were tokenized and truncated or padded to a maximum sequence length of 128 tokens. A linear classification head was added on top of the pooled <s> token representation for multi-class prediction. To address class imbalance, they applied weighted cross-entropy loss with manually computed class weights along with label smoothing (0.1). The model was fine-tuned using the AdamW optimizer, weight decay of 0.01, and a linear warmup over 10% of training steps, using mixed-precision (FP16) training.

5.8 cantnlp

This team (Li and Wong, 2026) used an approach of fine-tuning XLM-RoBERTa for both Task 1 and Task 2. They have used an external dataset namely "Corpus of Global Language Use" while training the model.

5.9 MUCS

The methodology proposed by the team MUCS (Shashirekha and A, 2026) extracted features including word n-grams, character n-grams, syllable n-grams and subword units using Byte Pair Encoding tokenizer. For Kannada, syllables are generated using Indic syllabifier while English syllables are generated using rule-based vowel grouping method. All extracted features are normalized and merged into a single feature representation. The combined features are transformed into numerical vectors using TF-IDF Vectorization. Several Machine Learning classifiers including MultinomialNB, Logistic Regression, K Nearest Neighbors, Decision Tree are trained on this representation. They used this methodology for both Task 1 and 2.

5.10 NamDang

The team NamDang participated only in Task 1 who employs the DSPy (Declarative Self-

improving Language Programs) framework to move beyond manual prompt engineering toward programmatic optimization. The core logic is defined through declarative signatures, which are then optimized using the BootstrapFewShot teleprompter. This process automatically curates and bootstraps high-quality demonstrations from the training data to guide the model's reasoning. Finally, a robust post-processing layer with fuzzy matching is used to ensure the Large Language Model's outputs are accurately mapped to the four required hope tone categories.

5.11 Prime_Line

Team PrimeLine (V et al., 2026) used XLM-RoBERTa by utilizing model's subword tokenizer and fine-tuned on the provided training data for a small number of epochs to avoid overfitting. This team used the same approach for both Task 1 and Task2.

Tables 6 and 7 summarizes the methodologies used by the teams for the tasks. Some of the teams such as CUET_Synthetica, SJM_MINDS and MUCS applied preprocessing techniques before training the models. A few teams such as CUET_Synthetica, SJM_MINDS and CUETEuropa employed techniques to handle data imbalance problems. One team "cantnlp" used external data set while training the model. Majority of the teams used XLM-RoBERTa to train the models. Team NamDang used prompting with few-shot approach in Task 1.

6 Results

The runs submitted by the teams are evaluated using the metrics namely accuracy, precision, recall and macro-F1 score values. Teams are ranked based on the macro-F1 obtained for their runs. The values for Task 1 are given in Table 8 and the values for Task 2 are provided in Table 9. It is observed that the team CUET_Synthetica who used ensemble of transformers secured the first position in Task 1. The same who fine tuned XLM-RoBERTa secured the top position in the leader board for Task 2.

7 Conclusion

This paper presented the findings of the Shared Task on Hope Speech Detection in Code-Mixed Tulu Language at DravidianLangTech@ACL 2026. It consists of 2 sub tasks namely Task 1: Coarse-Grained Hope Tone Classification and Task 2: Fine-Grained Hope Type Classification. Task 1 is a 4 class and Task 2 is a 5 class classification problems on hope speech identification. The task has 7504 instances for Task 1 and 4550 instances for Task 2. A total of 9 teams participated in both tasks. Additionally, 2 more teams participated only in Task 1. The team CUET_Synthetica secured first position in both the tasks. They used ensemble of transformers for Task 1 and fine tuned XLM-RoBERTa for Task 2. Though, some of the teams tried to address the data imbalancing problem, the methodology they used is not effective. Also, employing language agnostics embeddings may give a good results which was not tried by any of the team.

8 Limitations

The participating teams were able to achieve only 0.58 and 0.42 as the maximum scores for Task 1 and Task 2 respectively. Most of the teams employed traditional classifiers with bag of word features which do not capture any semantics to identify the hope speech. Other teams fine tuned transformers which is also not helped so much to improve the performance.

9 Ethical Considerations

The dataset used in this shared task is collected from social media postings that are publicly available by ensuring compliance with the terms and policies of social media platforms. Special care was taken to avoid collecting sensitive personal data related to religion, ethnicity, gender identity, health status, or political affiliation.

Acknowledgments

Author Bharathi Raja Chakravarthi was funded by a research grant from Research Ireland under grant number SFI/12/RC/2289_P2 (Insight). This work is also supported by the Centre for Research Training in Artificial Intelligence grant number 18/CRT/6223, and a grant from the College of Science and Engineering, University of Galway, Ireland.

References

Moogambigai A, Nandhika Saravanan, Nikhil Karthik S, Pandiarajan D, and Bharathi B. 2026. SSN_hopeNetters@DravidianLangTech 2026: Multi-level hope speech detection using XLM-RoBERTa. In *Proceedings of the Sixth Workshop on*

Speech, Vision, and Language Technologies for Dravidian Languages. Association for Computational Linguistics.

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. Borrowing or code-mixing? an analysis of english-hindi code-mixed social media text. *Proceedings of EMNLP Workshop on Computational Approaches to Code Switching*.

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 and Vigneshwaran Muralidaran. 2021. Findings of the shared task on hope speech detection for equality, diversity, and inclusion. In *Proceedings of the first workshop on language technology for equality, diversity and inclusion*, pages 61–72.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, D Thenmozhi, and John P McCrae. 2021. Hope speech detection in under-resourced languages. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH)*. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, and John P McCrae. 2020. Sentiment analysis for dravidian languages. In *Proceedings of FIRE*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, and et al. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*. Association for Computational Linguistics.

Sonith Divakaran, Kavya Girish, and Hosahalli Lakshmaiah Shashirekha. 2024. Hope on the horizon: Experiments with learning models for hope speech detection in spanish and english. In *IberLEF@ SEPLN*.

Joel Johnson, Meclin A Francis, Jyoti Kumari, Malavika Sreekumar, and Vinay Babu Ulli. 2026. Hope_Speech_Alchemists@DravidianLangTech 2026: TF-IDF SVM and XLM-RoBERTa with focal loss for hope speech detection in tulu. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, and et al. 2020. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of EMNLP*. Association for Computational Linguistics.

Klaus Krippendorff. 2018. *Content Analysis: An Introduction to Its Methodology*. Sage Publications.

Andrew Li and Sidney Wong. 2026. cantnlp@DravidianLangTech 2026: Organic domain adaptation improves multi-class hope speech detection in tulu. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Joyeta Barua Moni, Noore Tamanna Orny, Md. Abtahee Kabir, and Hasan Murad. 2026. Team Oryu@DravidianLangTech 2026: A multilingual transformer approach for hope speech detection in code-mixed tulu. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Hosahalli Lakshmaiah Shashirekha and Rachana A. 2026. MUCS@Dravidianlangtech@ACL2026: Hope speech detection in code-mixed tulu language using multiple features. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Hosahalli Lakshmaiah Shashirekha, Manjula Suvarna, and Jayashree Krishna. 2026. SJM_MINDS@DravidianLangTech@ACL 2026: Machine learning approaches for hope speech detection in code-mixed tulu. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Rithikaa V, Sanjay Krishnan K, Nithya Varshini C N R, and S. Sumathi. 2026. PrimeLine@DravidianLangTech 2026: Hope speech detection in tulu using XLM-RoBERTa for coarse and fine-grained classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Sumaiya Zaman, Miftahul Jannat Rishta, Shiti Chowdhury, and Hasan Murad. 2026. CUET_SYNTHETICA@DravidianLangTech 2026: Multilingual transformer based hope speech detection for coarse and fine-grained classification in tulu. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

From Comments to Harm: A Findings Report on Abusive Tamil Text Targeting Women on Social Media Shared Task

Bhuvaneswari Sivagnanam[1], Kathiravan Pannerselvam[1], Jananayagan V[1], Charmathi Rajkumar[2], Ramesh Kannan R[3], Ratnavel Rajalakshmi[3], Shunmuga Priya Muthusamy Chinnan[4], Saranya Rajiakodi[1], Bharathi Raja Chakravarthi[4]

[1]Central University of Tamil Nadu, India, [2]Kongu Engineering College, Erode, India
[3]Vellore Institute of Technology, Chennai, India,
[4]Unit for Inclusive AI, Data Science Institute, University of Galway, Ireland

Abstract

This paper presents an overview of the second shared task on Abusive Tamil Text Targeting Women on Social Media as a binary classification problem (abusive vs. non-abusive). We release a dataset of Tamil YouTube comments and evaluate submissions using macro-F1 to encourage balanced performance in a noisy, low-resource setting. There are 89 teams registered for this task and 24 teams submitted the results. The approaches used by the teams includes transformer fine-tuning, heterogeneous ensembles, classical baselines, and large language models using prompting and LoRA. Results show that the best-performing system scored 0.8297 macro-F1 and many submissions are around 0.79-0.81. Across submissions, transformer fine-tuning with domain-aligned encoders is consistently strong, while additional gains are frequently associated with Tamil-aware normalization and macro-F1-oriented calibration such as class-weighted learning and validation-based threshold tuning. Overall, the findings highlights the importance of language-aware preprocessing and careful decision calibration for reliable moderation of women-targeted abusive Tamil social media text.

Disclaimer: This paper (including figures and examples) may contain offensive or harmful language, including abusive content targeting women. All such text is presented solely for research and educational purposes and it does not reflect the author's views. Reader discretion is advised.

1 Introduction

In recent years, the social media platforms has grown significantly and reshaped how people communicate, share their views, and engage in public discussions (Ratkiewicz et al., 2011; Stephenson et al., 2018). While these digital spaces provide opportunities for connection and information sharing, they have also become platforms where harmful content, especially gender-based abuse, spreads (Suzor et al., 2019). Particularly, women are the most frequent targets of harassment, misogynistic comments and gender-based hostility within such online environments[1,2]. This not only affects their mental well-being and participation in digital platforms but also reflects broader social inequalities and discriminatory attitudes towards them (Kavanagh et al., 2019; Miranda, 2023).

In such settings, *offensive language* broadly refers to text that violates norms of respectful communication which includes insults, profanity, harassment, and other hostile expressions that can harm individuals or groups (Vidgen et al., 2019; Chakravarthi et al., 2021). *Abusive language* is more specific of offensive language in which the intent and effect are explicitly degrading, threatening, or humiliating toward a target (Vidgen et al., 2019) (Caselli et al., 2020). In this shared task, the target is women in which the abusive content frequently appears as misogynistic harassment, sexualised insults, or gendered ridicule aimed at policing women's presence and voice in public online spaces (Rajiakodi et al., 2025).

Automatically detecting such abuse is difficult even in high-resource languages because many abusive comments are indirect as they rely on sarcasm, insinuation, stereotypes, or context-dependent meanings so they cannot be captured reliably by simple lexical matching (Ocampo et al., 2023; Vidgen et al., 2019). The challenge is increased for Tamil, a low-resource language, where social-media writing introduces additional variability (Pannerselvam and Rajiakodi, 2026; Rajiakodi et al., 2025) such as informal and non-standard spellings, spoken forms, and regional variation, along with pervasive Tamil-English code-mixing

[1] https://www.amnesty.org/en/documents/act30/8070/2018/en/

[2] https://www.unwomen.org/en/articles/faqs/digital-abuse-trolling-stalking-and-other-forms-of-technology-facilitated-violence-against-women

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 88–100
July 3, 2026 ©2026 Association for Computational Linguistics

and Romanised writing (Chakravarthi et al., 2021; Saumya et al., 2021; Prasanna and Arora, 2024). These characteristics increase lexical sparsity and complicate tokenisation and feature learning, reducing the robustness of conventional text-processing pipelines (Saumya et al., 2021; Prasanna and Arora, 2024). As a result, building reliable models for women-targeted abusive Tamil detection remains an active research problem, motivating shared-task benchmarks that enable systematic comparison of approaches under a unified evaluation protocol (Rajiakodi et al., 2025; Seemann et al., 2023).

Content moderation increasingly combines encoder-based classifiers with large language models (LLMs). DetoxBench reports that LLM (e.g., GPT-style models, Claude, Mistral/Mixtral, and Jurassic) performance varies widely across abuse categories (Chakraborty et al., 2024), and studies on implicit abuse show that zero-/few-shot prompting can help but remains inconsistent for context-dependent cases (Jaremko et al., 2025). This shared task builds on the first shared task on abusive Tamil and Malayalam text targeting women (Rajiakodi et al., 2025). We organize the *second* edition, *Abusive Tamil Text Targeting Women on Social Media*, as part of DravidianLangTech@ACL 2026[3], with a slightly modified setup to support clearer benchmarking and analysis. The objective of this shared task was to encourage the participants to develop a models capable of detecting abusive text targeting women in social media.

This paper is structured as follows: we describe the dataset construction process and annotation guidelines, then summarize the participating systems and evaluation setup. Next, we report the official results and highlight key observations and trends across submissions. Finally, we discuss common challenges and error patterns, and outline directions for future work toward more reliable and inclusive moderation for Tamil-speaking online communities.

2 Related work

Research on abusive text classification advanced by the several shared tasks and datasets. Hate Speech (Satapara et al., 2023; Mubarak et al., 2022) and offensive content detection (Chakravarthi et al., 2022) shared task on English and other Indian languages advanced the deeper research on abusive text classification. The Hasoc (Satapara et al., 2023; Mubarak et al., 2022) shared task adopted twitter dataset for English, Arabic and Indo-Aryan Languages for hate speech detection. The DravidianCodeMix (Chakravarthi et al., 2022) dataset provided 60000 instances for offensive language detection on code-mixed languages of Tamil-English, Malayalam-English, and Kannada-English. Recently Hate speech identification on Bangala language (Hossan et al., 2025) adopted multi-task setup to include the severity and target group identification. These shared tasks established the benchmarking procedures and standard datasets for abusive language detections.

English is a high-resource language, extensive studies have been carried (Fortuna et al., 2021) for identifying Abusive language, but limited research on low-resource languages such as Tamil. Research on languages such as Tigrinya (Gaim et al., 2025), German (Satapara et al., 2023) and various Indian languages underscores the importance of developing annotated datasets and multilingual models to address the resource scarcity. Recent research on code-mixed datasets for languages such as Telugu-English and Nepali-English (Pandey et al., 2025; Zia Ur Rehman et al., 2023) introduced to support multilingual abusive language detection. These works highlights the significance of multilingual embeddings and cross-lingual transfer learning in developing abusive language detection in low-resource settings.

Abusive text classification evaluation practices rely on standard metrics such as accuracy, precision, recall, and F1-score (Davidson et al., 2017; Ganganwar and Rajalakshmi, 2022). Shared tasks on abusive comment detection in various languages has commonly used macro F1-score (Fortuna et al., 2021; Satapara et al., 2024) as the primary evaluation metric due to class imbalance issues. Binary classification schemes (Priyadharshini et al., 2023) and Cross-dataset evaluation (Fortuna et al., 2021) have been adopted across domains to test the model generalization. Further studies (Chiril et al., 2022) shows statistical validation techniques such as validation and significance testing to ensure reliable performance comparisons.

The study of gender-targeted abuse has emerged as a key area within abusive language detection. Shared tasks and various datasets focusing on regional languages and English have been developed to identify the targeted women using abusive contents and targeted on marginalized groups.

[3] https://sites.google.com/view/dravidianlangtech-2026/home?authuser=0

Hate speech datasets such as HatEval (Basile et al., 2019), (Chiril et al., 2022) explicitly added women as a target group for hate speech identification. Another major challenge is the detection of abusive content in code-mixed and Romanized text such as Tanglish (Benhur and Sivanraju, 2021). Transformer-based models have been successfully applied to Tanglish offensive language detection tasks using multilingual pretrained representations. Code-mixed datasets such as DravidianCodeMix (Chakravarthi et al., 2022) further demonstrate the linguistic complexity of Romanized and multilingual social media text. Intimate Partner Violence (IPV) related posts were detected using a RoBERTa based transformer model approach on manually annotated English dataset collected from Twitter and Reddit (Guo et al., 2023). The model able to distinguish abusive self reports from general discussions with an accuracy of 78 %. The dataset is small and it is limited to detect explicit English comments. These challenges highlight the urgent need for robust models capable of handling low resource, multilingual and noisy texts in social media. All the above discussed articles explore various implementations and performance metrics for abusive text detection, hate speech detection and offensive language detection. They provide valuable insights into implementation strategies and deep learning models. However, there remains a critical gap in targeting gender based abuse towards women in particularly Dravidian language like Tamil.

3 Task description

This second edition of the shared task is organized on the Codabench platform[4] and focuses on binary classification of Tamil YouTube comments. In this edition, we only detect abusive comments that target women. Each comment is classified as either Abusive or Non-Abusive.

A comment is labeled as Abusive only if the abusive content is targeted at women. Comments that contain bad words but do not target women are labeled as Non-Abusive. For example, if a man is insulted using offensive words, even if those words mention his mother or other female family members, the comment is still considered Non-Abusive because the main target is not a woman. Most abusive comments targeting women include references to female body parts or insults about women's character, behavior, or morality.

Further, the abusive category includes comments that contain offensive, insulting, threatening, harassing or otherwise harmful language. The abuse may be explicit or implicit, and may appear in direct, sarcastic, or coded forms, but it reflects an intention to demean or harm the target, and the Non-Abusive category includes comments that are neutral, positive, informative, or express general opinions without harmful intent. To support system development and evaluation, annotated datasets are provided in Tamil, including training, and test splits. Participants are required to build models that automatically classify each test comment into one of the two predefined categories. The task aims to advance research on the detection of abusive Tamil language and to contribute to safer online communication in low-resource language settings.

4 Dataset

We created a Tamil YouTube comment dataset with annotations for abusive text targeting women. The comments were collected using the automated scraping tools combined with manual filtering from publicly available comment sections of popular Tamil YouTube channels. Comments were gathered from publicly available videos related to entertainment, politics, and social discussions to make sure diversity in abusive and non-abusive expressions. The dataset contains abusive comments specifically directed at women, as well as comments that are not targeted at women, even if they target men or are neutral. It was prepared to reflect normal social media Tamil language usage, including informal spelling, slang words, and platform-style punctuation. The classification task is binary, with two labels: Abusive (targeting women) and Non-Abusive (not targeting women).

4.1 Annotation process

Annotator background. The annotation process was done by three annotators with different academic backgrounds to improve labeling quality. As shown in Table 1, two annotators were Computer Science students and one was a Social Work student. Among them, two were Ph.D. students and one was a postgraduate student. This diversity helped reduce bias when identifying abusive content, since interpreting abusive language can sometimes be subjective.

All annotators were native Tamil speakers and

[4] https://www.codabench.org/competitions/11326/#/pages-tab

were familiar with informal social media language, including slang words and casual writing styles. Before annotation, they were given clear guidelines explaining abusive content targeting women, including both direct and indirect abusive expressions. When there were differences in opinions, the annotators discussed and agreed on a final label. This helped improve the consistency and quality of the dataset labels.

Characteristic	Details
# Annotators	3
Gender Distribution	2 Female, 1 Male
Educational Background	2 CS Students 1 SW Student
Level of Study	2 Ph.D. and 1 PG
Native Language	Tamil
Social Media Language Familiarity	Yes

Table 1: Annotator Demographic Details. CS- Computer Science, SW- Social Work

Guidelines and labeling. Annotators labeled each comment based on whether it contained abusive content targeting women. The interpretation of Abusive and Non-Abusive content can sometimes be subjective, as it depends on social and cultural context. Therefore, multiple annotators evaluated each comment to improve labeling reliability. Direct abusive content included insults, derogatory words, or explicit attacks targeting women. Contextual meaning was also considered during annotation. Figure 1 and Figure 2 present sample annotated comments.

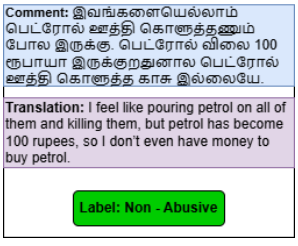


Figure 1: Example of Non-Abusive Comment Expressing Anger. Although the comment expresses emotional anger, it does not directly contain abusive language targeting women.

Each comment was annotated by multiple annotators, and the final label was determined using majority voting. This approach helped reduce individual bias and improved labeling consistency, especially for borderline cases where subjective interpretation may vary.

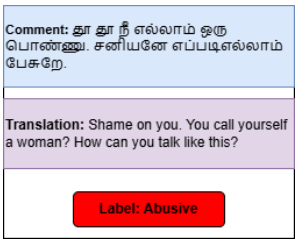


Figure 2: Example of Abusive Comment. The comment contains direct insulting expressions targeting women and was therefore labeled as abusive.

Split	#Abusive	#Non-Abusive	Total
Train	1769	1883	3652
Test	441	472	913
Total	2210	2355	4565

Table 2: Label distribution of the dataset.

Inter-annotator agreement. For this task, the Krippendorff's Alpha (Marzi et al., 2024) is 0.6474, indicating moderate agreement. This level of agreement is common in subjective social-media moderation tasks where sarcasm, implicit harassment, and borderline insults can lead to genuine annotator disagreement.

Dataset splits and evaluation setup. The dataset is split into a training set (3652 instances) and a test set (913 instances) (Table 2). The test set is released in two forms: an unlabeled version for evaluation and a labeled version for subsequent error analysis.

5 Evaluation setup

The submitted models are evaluated using standard classification metrics computed over the two classes: *Abusive* and *Non-Abusive*. The performance of the classification system is measured using Accuracy, Precision, Recall, macro F1-score, and Weighted F1-score (Sharma et al., 2024).

Accuracy measures the proportion of correctly predicted instances among the total number of instances. Precision measures the proportion of correctly predicted instances of a class among all instances predicted for that class. Recall measures the proportion of correctly identified instances of a class among all actual instances of that class. The F1-score combines Precision and Recall into a single measure by taking their balanced mean.

The primary evaluation metric for ranking is the macro-averaged F1-score. The F1-score is first computed independently for each class. The macro F1 score (Eq. (1)) is then obtained by averaging the

F1-scores of all classes, giving equal importance to each class regardless of its size (Hinojosa Lee et al., 2024).

$$\text{macro F1} = \frac{1}{N}\sum_{i=1}^{N} F1_i \tag{1}$$

Since abusive language datasets are often imbalanced, the macro F1-score ensures that performance on the minority class is properly reflected. The final ranking of systems is based on the macro-averaged F1-score computed on the test set.

Weighted F1-score (Eq. (2)) computes the F1-score for each class and averages them according to the proportion of instances in each class.

$$Weighted\ F1 = \sum_{i=1}^{N} \left(\frac{n_i}{N} \times F1_i\right) \tag{2}$$

where N represents the total number of classes, and n_i denotes the number of instances in class i.

Participants are allowed to submit multiple runs. All valid submissions are considered and the highest macro-F1 score for each participant is used to determine the final ranking. Only the best-performing submission from each participant is included in the official results. The use of external data and pre-trained models is permitted. However, participants must clearly describe any additional resources used in their system description paper to ensure transparency. Participants must submit their predictions for the test set. Each line in the submission file should correspond to a test instance and contain the predicted label (Abusive or Non-Abusive) in the same order as the test data. Submissions that do not follow the required format may not be evaluated. This evaluation setup ensures fairness and clarity in comparing different models.

6 Overview of participant systems

The submitted systems span diverse approaches for detecting abusive Tamil text targeting women on social media, ranging from fine-tuned transformer encoders to ensembles, LLM-based prompting/LoRA, and lightweight classical baselines. Several submissions also reported robustness-oriented preprocessing for noisy or obfuscated text, and a small subset used external data.

6.1 Top-ranked systems (by macro-F1)

The top of the leaderboard is dominated by transformer-based systems with either domain adaptation, Indian language pretraining, or Tamil-aware normalization. nitc-hsr (Rank 1) submitted an ensemble of Indian-language/domain-adapted encoders (IndicBERT and AbuseXLMR). primeline_abusive_Tamil (V et al., 2026) (Rank 2) fine-tuned MuRIL using a simple and reproducible pipeline without handcrafted features. HNK (Rank 3) emphasized grapheme-aware preprocessing before tokenization and reported consistent gains across multiple transformer backbones. CUET_Synthetica (Rank 4) proposed a large heterogeneous ensemble combining Tamil-specialized models (e.g., MuRIL, IndicBERTv2) with multilingual transformers (e.g., XLM-RoBERTa-Large), together with duplicate removal and Tamil-specific preprocessing. DPR (Prakash et al., 2026) (Rank 5) conducted a controlled comparison of multilingual transformers (mBERT, MuRIL, XLM-RoBERTa, IndicBERT) under consistent settings to study transfer robustness for Tamil social-media abuse detection.

6.2 Transformer-based fine-tuning

Many teams contributed strong baselines by varying pretraining choice and fine-tuning configuration. CHMODE_777 (Karunanidhi and Arumugam, 2026) compared MuRIL and XLM-RoBERTa, highlighting the importance of model selection and reporting gains from longer context windows (up to 256 tokens) and Indian language pretraining. DLRG (Sankar and Rajalakshmi, 2026) fine-tuned MuRIL with label normalization and HTML cleaning, using stratified splits, mixed-precision training, weight decay, and warmup scheduling; they emphasised that carefully optimised fine-tuning can be a strong baseline without external data or ensembles. IndiLangTech fine-tuned IndicBERTv2 with a standard classification head. TriVector (Rebayet et al., 2026) used a Tamil hate-speech domain-adapted XLM-R model and incorporated statistical lexicon features. ADAPTIVEMINDS fine-tuned XLM-RoBERTa directly on the provided labels with a lightweight setup, without manual relabeling/normalization. Medhastra_abusive also fine-tuned XLM-RoBERTa and highlighted a straightforward end-to-end pipeline without rules, lexicons, or external resources.

6.3 Ensembles submissions

A subset of teams explored ensembles and hybrid pipelines to improve robustness. Lannisters reported Tamil-aware normalization (Unicode/HTML handling), explicit handling of obfuscated/censored abuse, lightweight augmentation/noise strategies, and confidence-aware prediction for flagging low-confidence cases. SuperNova (K et al., 2026) evaluated multiple paradigms, including end-to-end MuRIL fine-tuning and embedding-based classifiers (MuRIL or SBERT embeddings with tree/boosting models), highlighting trade-offs between accuracy and simplicity.

6.4 macro-F1-based optimization

Several submissions explicitly targeted balanced performance using macro-F1-aware optimization choices. N-cuboid-MP applied class-weighted training and validation-based threshold optimization rather than using a fixed 0.5 decision boundary. DynamicDuo emphasized cost-sensitive fine-tuning and probability-based decision thresholding to reduce false negatives.

6.5 LLM-based approaches and deployment-oriented framing

Two submissions explicitly connected the task to modern moderation workflows using LLMs or human-in-the-loop design. VITECH (Krubhakaran et al., 2026) evaluated multiple LLM families (Gemma, LLaMA, Qwen, and DeepSeek-Distilled) under zero-shot/few-shot prompting and LoRA fine-tuning, reporting that LoRA and larger model sizes (e.g., 8B) consistently improved performance over smaller models. Infinity_Preetha proposed a deployable moderation workflow layered on top of transformer classification, using a progressive three-warning escalation mechanism that forwards persistent offenders to human moderators for verification and action.

6.6 Lightweight baselines and external data

A few teams reported lightweight pipelines that remain attractive for efficiency and interpretability. GauriWarriors used a FastText-based approach with subword modeling and auto-tuning to handle spelling variation and informal text. KEC's Code Crafters used an external Kaggle dataset[5] and combined word- and character-level features in a simple ML classifier to capture noisy spelling patterns. KECInference submitted two contrasting runs like MuRIL fine-tuning and a TF-IDF+SVM baseline to illustrate the trade-off between contextual encoders and lexical models. Finally, tamilgoodbadtxt presented a model-agnostic transformer pipeline enabling consistent preprocessing and easy switching across pretrained backbones, while VISION (Subramanian et al., 2026) and CodeTamizh described practical pipelines emphasizing preprocessing and scalability, although the exact backbone and settings were not fully specified in their short submission notes.

[5] https://www.kaggle.com/datasets/prfsr007/abusive-tamil-dataset

7 Results and Discussion

There are total of 89 teams registered for this shared task, and among them 24 teams submitted their results for ranking and evaluation. This gap may due to the fact of practical challenges associated with the low-resource Tamil abusive language detection which includes the noisy social media text and computational constraints.Table 3 reports the official leaderboard for the shared task which ranks the systems by macro-F1 (our primary evaluation metric). Alongside macro-F1, we also reported Accuracy, Precision, Recall, and Weighted-F1 for completeness.

7.1 Overall leaderboard

The top system, nitc-hsr (A and Kumar, 2026) (Run 2), achieved the best overall performance with a macro-F1 of 0.8297, followed by prime-line_abusive_Tamil (V et al., 2026) (Run 1, macro-F1 0.8133) and HNK (R et al., 2026) (Run 1, macro-F1 0.8103). The next positions were occupied by CUET_Synthetica (Rishta et al., 2026) (Run 2, 0.8086) and DPR (Prakash et al., 2026) (Run 4, 0.8072), indicating a competitive top tier where multiple transformer-based systems performed within a narrow margin. Several teams achieved macro-F1 values are around 0.79-0.80 which demonstrates that strong performance is achievable under a range of design choices.

7.2 Statistical Significance Analysis

To examine whether the differences among the top-ranked systems were statistically meaningful, we performed bootstrap significance testing with 10,000 resampling iterations using Macro-F1 as the evaluation metric. The analysis was conducted on the prediction outputs of the top three systems. The

Team	RUN	Accuracy	Precision	Recall	macro-F1	Weighted-F1	Rank
nitc-hsr (A and Kumar, 2026)	RUN 2	0.8302	0.8310	0.8293	0.8297	0.8300	1
primeline_abusive_Tamil (V et al., 2026)	RUN 1	0.8138	0.8140	0.8131	0.8133	0.8100	2
HNK (R et al., 2026)	RUN 1	0.8105	0.8103	0.8104	0.8103	0.8000	3
CUET_Synthetica (Rishta et al., 2026)	RUN 2	0.8094	0.8106	0.8082	0.8086	0.8100	4
DPR (Prakash et al., 2026)	RUN 4	0.8072	0.8072	0.8076	0.8072	0.8100	5
CHMODE_777 (Karunanidhi and Arumugam, 2026)	RUN 2	0.8061	0.8061	0.8065	0.8061	0.8000	6
Infinity_Preetha	RUN 4	0.8061	0.8059	0.8059	0.8059	0.8100	7
Lannisters	RUN 1	0.8028	0.8032	0.8020	0.8023	0.8000	8
VISION (Subramanian et al., 2026)	RUN 10	0.8018	0.8018	0.8022	0.8017	0.6800	9
N-Cuboid-MP	RUN 2	0.8007	0.8010	0.7998	0.8001	0.8000	10
SuperNova (K et al., 2026)	RUN 1	0.8007	0.8031	0.7989	0.7994	0.6800	11
VITECH (Krubhakaran et al., 2026)	RUN 1	0.7963	0.7963	0.7957	0.7959	0.8000	12
IndiLangTech	RUN 1	0.7941	0.7938	0.7940	0.7939	0.7900	13
DLRG (Sankar and Rajalakshmi, 2026)	RUN 1	0.7919	0.7919	0.7913	0.7915	0.7900	14
TriVector (Rebayet et al., 2026)	RUN 1	0.7897	0.7895	0.7898	0.7896	0.7900	15
tamilgoodbadtxt (K and B, 2026)	RUN 1	0.7864	0.7881	0.7876	0.7864	0.7800	16
ADAPTIVEMINDS	RUN 1	0.7700	0.7735	0.7678	0.7681	0.7700	17
GauriWarriors	RUN 1	0.7338	0.7333	0.7010	0.7117	0.7335	18
CodeTamizh	RUN 1	0.5268	0.5246	0.5181	0.4881	0.6800	19
Medhastra_abusive	RUN 1	0.5170	0.2585	0.5000	0.3408	0.6800	20
DynamicDuo	RUN 1	0.5170	0.2585	0.5000	0.3408	0.6800	20
DravidianNLP	RUN 1	0.5170	0.2585	0.5000	0.3408	0.6800	20
KEC's Code Crafters	RUN 1	0.5170	0.2585	0.5000	0.3408	0.6800	20
KECInference	RUN 1	0.5170	0.2585	0.5000	0.3408	0.6800	20

Table 3: Leaderboard of Participating Systems Ranked by macro-F1

Team	Model(s) reported	Pre proc	Indian-PT	Ens-emble	Ext. Data	Imb.	Thr.	Aug.	HITL	LLM/ LoRA
nitc-hsr (A and Kumar, 2026)	IndicBERT + AbuseXLMR	✗	✓	✓	✗	✗	✗	✗	✗	✗
primeline_abusive_Tamil (V et al., 2026)	MuRIL	✗	✓	✗	✗	✗	✗	✗	✗	✗
HNK (R et al., 2026)	Multiple transformers + grapheme-aware normalization	✓	✓	✗	✗	✗	✗	✗	✗	✗
CUET_Synthetica (Rishta et al., 2026)	MuRIL, IndicBERTv2, XLM-R	✓	✓	✓	✗	✗	✗	✗	✗	✗
DPR (Prakash et al., 2026)	mBERT, MuRIL, XLM-R, IndicBERT	✗	✓	✗	✗	✗	✗	✗	✗	✗
CHMODE_ 777	MuRIL vs XLM-R	✗	✓	✗	✗	✗	✗	✗	✗	✗
Infinity_Preetha	XLM-R + progressive 3-warning escalation workflow	✗	✗	✗	✗	✗	✗	✗	✓	✗
Lannisters	XLM-R	✓	✗	✗	✗	✗	✓	✓	✓	✗
VISION (Subramanian et al., 2026)	(Not specified)	✓	✗	✗	✗	✓	✗	✗	✗	✗
N-Cuboid-MP	XLM-R fine-tuning	✗	✗	✗	✗	✓	✓	✗	✗	✗
SuperNova (K et al., 2026)	MuRIL FT	✗	✓	✓	✗	✗	✗	✗	✗	✗
VITECH (Krubhakaran et al., 2026)	Gemma / LLaMA / Qwen / DeepSeek	✗	✗	✗	✗	✗	✗	✗	✗	✓
IndiLangTech	IndicBERTv2 fine-tuning	✗	✓	✗	✗	✗	✗	✗	✗	✗
DLRG (Sankar and Rajalakshmi, 2026)	MuRIL fine-tuning	✓	✓	✗	✗	✗	✗	✗	✗	✗
TriVector (Rebayet et al., 2026)	Hate-speech-CNERG/deoffxlmr-mono-tamil	✗	✗	✗	✗	✗	✗	✗	✗	✗
tamilgoodbadtxt (K and B, 2026)	Model-agnostic transformer pipeline	✓	✓	✗	✗	✗	✗	✗	✗	✗
ADAPTIVEMINDS	XLM-R fine-tuning	✗	✗	✗	✗	✗	✗	✗	✗	✗
GauriWarriors	FastText	✗	✗	✗	✗	✗	✗	✗	✗	✗
CodeTamizh	Model not specified	✗	✗	✗	✗	✗	✗	✗	✗	✗
Medhastra_abusive	XLM-R fine-tuning	✗	✗	✗	✗	✗	✗	✗	✗	✗
DynamicDuo	XLM-R	✗	✗	✗	✗	✓	✓	✗	✗	✗
DravidianNLP	(Not specified)	✓	✗	✗	✗	✗	✗	✗	✗	✗
KEC's Code Crafters	Traditional ML	✗	✗	✗	✓	✗	✗	✗	✗	✗
KECInference	MuRIL and TF-IDF + SVM	✗	✓	✗	✗	✗	✗	✗	✗	✗

Table 4: Summary of Descriptive trends of Participating Systems: Reported models and Techniques (✓ = reported, ✗ = not reported in the team description). **Preproc**: cleaning/normalization; **Indian-PT**: Indian-language pretraining (e.g., MuRIL/IndicBERT/AbuseXLMR); **Imb.**: class-imbalance handling (e.g., weighted loss/cost-sensitive); **Thr.**: validation-based threshold tuning; **Aug.**: data augmentation; **HITL**: human-in-the-loop moderation workflow; **LLM/LoRA**: prompting or LoRA fine-tuning of LLMs.

results (5) show that although nitc-hsr (A and Kumar, 2026) achieved the highest Macro-F1 score, its improvement over primeline_abusive_Tamil (V et al., 2026) was not statistically significant. Similarly, primeline_abusive_Tamil (V et al., 2026) and HNK (R et al., 2026) showed highly overlapping performance. nitc-hsr (A and Kumar, 2026) showed only a marginal advantage over HNK (R et al., 2026). These findings suggest that the top-ranked systems achieved broadly comparable performance, and small leaderboard differences should be interpreted cautiously.

Comparison	Mean Δ F1	95% CI	p-value
nitc-hsr (A and Kumar, 2026) vs PRIMELINE	0.0164	[-0.0047, 0.0376]	0.0676
nitc-hsr (A and Kumar, 2026) vs HNK (R et al., 2026)	0.0192	[-0.0015, 0.0400]	0.0365
PRIMELINE vs HNK (R et al., 2026)	0.0028	[-0.0178, 0.0237]	0.3992

Table 5: Bootstrap significance testing among the top-ranked systems using Macro-F1.

7.3 Trends across submissions

To avoid relying only on qualitative descriptions, we summarize reported techniques using Table 4. Overall, 10/24 submissions used Indian language or domain-adapted pretrained encoders (e.g., MuRIL/IndicBERT/AbuseXLMR), and 6/24 explicitly mentioned Tamil-aware preprocessing or

normalization (e.g., grapheme/Unicode handling, HTML/obfuscation cleaning). This pattern aligns with the leaderboard, where higher-ranked systems frequently adopted language/domain-aligned pretraining and, in some cases, Tamil-specific normalization.

A smaller subset of teams explicitly targeted macro-F1 through imbalance-aware optimization, such as class-weighted training, cost-sensitive loss functions, or validation-based threshold tuning. While the class distribution is relatively balanced in our dataset, such techniques are commonly used to calibrate decision boundaries and improve class-balanced metrics. Ensembles were explored by several teams.

LLM-based approaches were evaluated as moderation-relevant baselines using prompting (zero-shot/few-shot) and parameter-efficient adaptation (LoRA). While such methods offer flexibility, encoder-based fine-tuning remains a strong and computationally efficient choice for this binary classification setting.

7.4 Discussions

From the leaderboard and the reported system designs, we highlight three observations. First, reported gains frequently came from language/domain-aligned pretraining and Tamil-aware normalization, which help models handle noisy social-media text. Second, calibration choices including class weighting and validation-based threshold tuning can improve macro-F1 by producing more balanced predictions. Third, robustness-oriented preprocessing (e.g., handling obfuscated or masked abuse) is better aligned with real-world moderation needs.

Finally, multiple teams obtained identical macro-F1 values at the bottom of the leaderboard (Rank 20). To better understand this pattern, we examined the prediction distributions of several lower-ranked systems. The analysis showed that these systems did not uniformly collapse to a single class prediction; instead, they predicted both *Abusive* and *Non-Abusive* labels, although with varying degrees of bias toward the majority class. These observations suggest that identical macro-F1 scores at the bottom of the leaderboard may arise from different failure modes, including majority-class bias, skewed decision thresholds, difficulty handling minority-class instances, and inconsistent prediction behavior, rather than complete majority-class collapse alone.

8 Conclusion

This shared task introduced a benchmark for detecting abusive Tamil text targeting women in social-media comments that often contain code-mixing, informal spelling, and non-standard orthography. A total of participating teams explored a range of approaches, including Tamil-aware preprocessing, Indic-language pretrained transformers, and imbalance-aware training strategies. The best-performing system achieved a macro-F1 score of 0.8297, while several systems obtained competitive results in the 0.7900-0.8100 range. These results demonstrate that language-specific preprocessing and robust training techniques can improve abusive language detection in noisy online environments. However, several challenging cases remain, particularly those involving masked abusive words, ambiguous targets, quoted harmful language, and sarcasm. The shared task dataset and benchmark are expected to support further research on abusive language detection for low-resource Dravidian languages. Future work can explore finer-grained annotation schemes, contextual modeling, and cross-platform evaluation to improve robustness and generalization.

9 Limitations

The dataset used in this shared task is collected from YouTube comments, which may introduce platform-specific communication patterns and topical biases. As a result, models trained on this dataset may not directly generalize to other social media platforms without additional adaptation (Vidgen and Derczynski, 2020). Tamil social-media text also contains dialectal variation, transliteration, creative spelling, and frequent English code-mixing, which makes normalization and interpretation challenging. Another limitation concerns the binary labeling scheme, which compresses multiple forms of harmful language into a single category and may fail to capture implicit abuse, sarcasm, or counterspeech. In addition, the distinction between abusive and non-abusive comments can sometimes be subjective. During the shared task, some participants over mail sought clarification regarding this boundary, particularly in cases involving sarcasm, indirect criticism, or quoted language. This limitation may be addressed in future shared tasks through finer-grained annotation schemes. Finally, the limited availability of large-scale Tamil resources, including domain-pretrained models and

extensive annotated corpora, may restrict achievable system performance.

Ethical Statement

This shared task dataset contains abusive Tamil social media text targeting women and may include offensive or distressing language, it is provided strictly for research purposes and does not reflect the author's views. We sourced text from publicly available content, removed direct personal identifiers where possible, and released only the minimum information needed for the task, with annotation conducted by trained Tamil-proficient annotators under content warnings and well-being safeguards. The dataset is intended to support research on responsible moderation, but it may carry risks of bias and misuse, we therefore recommend careful, transparent use with human oversight and evaluation of potential harms.

Usage of AI Declaration

We used AI-based tools to support language editing and formatting of this manuscript (e.g., improving clarity, grammar, and LaTeX presentation). All technical content, experimental results, interpretations, and final wording were reviewed and verified by the authors. No AI system was used to generate or alter the dataset, annotations, or official leaderboard results.

Acknowledgements

The authors, Bharathi Raja Chakravarthi was funded by a research grant from Research Ireland under grant number SFI/12/RC/2289_P2 (Insight_2).

References

Rameez Mohammed A and S D Madhu Kumar. 2026. NITC-HSR@DravidianLangTech 2026: Ensembling Multilingual Transformer Models for Detecting Abusive Tamil Text Targeting Women on Social Media. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*, pages 54–63.

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

Tommaso Caselli, Valerio Basile, Jelena Mitrović, Inga Kartoziya, and Michael Granitzer. 2020. I feel offended, don't be abusive! implicit/explicit messages in offensive and abusive language. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6193–6202, Marseille, France. European Language Resources Association.

Joymallya Chakraborty, Wei Xia, Anirban Majumder, Dan Ma, Walid Chaabene, and Naveed Janvekar. 2024. Detoxbench: Benchmarking large language models for multitask fraud & abuse detection. *arXiv preprint arXiv:2409.06072*.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Navya Jose, Thomas Mandl, Prasanna Kumar Kumaresan, Rahul Ponnusamy, John Philip McCrae, Elizabeth Sherly, and 1 others. 2021. Findings of the shared task on offensive language identification in tamil, malayalam, and kannada. In *Proceedings of the first workshop on speech and language technologies for Dravidian languages*, pages 133–145.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Navya Jose, Elizabeth Sherly, and John McCrae. 2022. Dravidiancodemix: Sentiment analysis and offensive language identification dataset for dravidian languages in code-mixed text. *Language Resources and Evaluation*.

Patricia Chiril, Endang Pamungkas, Farah Benamara, Véronique Moriceau, and Viviana Patti. 2022. Emotionally informed hate speech detection: A multi-target perspective. *Cognitive Computation*, 14(3):1322–1336.

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

Paula Fortuna, Juan Soler-Company, and Leo Wanner. 2021. How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets? *Information Processing & Management*, 58(3):102524.

Fitsum Gaim, Hoyun Song, Huije Lee, Changgeon Ko, Eui Jun Hwang, and Jong C. Park. 2025. A multitask benchmark for abusive language detection in low-resource settings. *Preprint*, arXiv:2505.12116.

Vaishali Ganganwar and Ratnavel Rajalakshmi. 2022. Mtdot: A multilingual translation-based data augmentation technique for offensive content identification in tamil text data. *Electronics*, 11(21).

Yuting Guo, Sangmi Kim, Elise Warren, Yuan-Chi Yang, Sahithi Lakamana, and Abeed Sarker. 2023. Automatic detection of intimate partner violence victims from social media for proactive delivery of support.

In *AMIA Joint Summits on Translational Science Proceedings*, pages 254–260. American Medical Informatics Association.

Maria Cristina Hinojosa Lee, Johan Braet, and Johan Springael. 2024. Performance metrics for multilabel emotion classification: comparing micro, macro, and weighted f1-scores. *Applied Sciences*, 14(21):9863.

Md. Refaj Hossan, Kawsar Ahmed, and Mohammed Moshiul Hoque. 2025. CUET-NLP_Zenith at BLP-2025 task 1: A multi-task ensemble approach for detecting hate speech in Bengali YouTube comments. In *Proceedings of the Second Workshop on Bangla Language Processing (BLP-2025)*, pages 443–452, Mumbai, India. Association for Computational Linguistics.

Julia Jaremko, Dagmar Gromann, and Michael Wiegand. 2025. Revisiting implicitly abusive language detection: Evaluating llms in zero-shot and few-shot settings. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3879–3898.

Kiruthika K, Roahiyaa T, and Premjith B. 2026. SUPERNOVA@DravidianLangTech 2026: Transformer and Ensemble Approaches for Abusive Tamil Text Detection Targeting Women. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Varalakshmi K and Bharathi B. 2026. TAMILGOODBADTXT@DravidianLangTech 2026:A Multilingual Transformer-Based Approach for Abusive Language Identification in Tamil Social Media. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Arunaggiri Pandian Karunanidhi and Prabalakshmi Arumugam. 2026. CHMOD_777@DravidianLangTech 2026: Context-Aware Fine-tuned MuRIL for Abusive Tamil Text Detection on Social Media. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Emma Kavanagh, Chelsea Litchfield, and Jaquelyn Osborne. 2019. Sporting women and social media: Sexualization, misogyny, and gender-based violence in online spaces. *International Journal of Sport Communication*, 12(4):552–572.

Triambiga Krubhakaran, Senthil Kumar B, Kaviya Nagarajan, and Balaji N. 2026. VITECH@DravidianLangTech2026: Prompting and LoRA Adaptation for Tamil Abusive Language Detection - A Comparative Study of Open LLMs. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Giacomo Marzi, Marco Balzano, and Davide Marchiori. 2024. K-alpha calculator–krippendorff's alpha calculator: a user-friendly tool for computing krippendorff's alpha inter-rater reliability coefficient. *MethodsX*, 12:102545.

Sandra Lopes Miranda. 2023. Analyzing hate speech against women on instagram. *Open Information Science*, 7(1):20220161.

Hamdy Mubarak, Hend Al-Khalifa, and Abdulmohsen Al-Thubaity. 2022. Overview of osact5 shared task on arabic offensive language and hate speech detection. In *OSACT*.

Nicolás Benjamín Ocampo, Ekaterina Sviridova, Elena Cabrio, and Serena Villata. 2023. An in-depth analysis of implicit and subtle hate speech messages. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1997–2013.

Manish Pandey, Nageshwar Prasad Yadav, Mokshada Adduru, and Sawan Rai. 2025. Creating and evaluating code-mixed nepali-english and telugu-english datasets for abusive language detection using traditional and deep learning models. *Preprint*, arXiv:2504.21026.

Kathiravan Pannerselvam and Saranya Rajiakodi. 2026. Systematic literature review on hate speech detection in indian low-resource languages. *Journal of Computational Social Science*, 9(1):5.

Diya Prakash, Praveen Kumar S, Ranjith Kumar R, Siranjeevi Rajamanickam, Balasubramanian Palani, and Jobin Jose. 2026. DPR@DravidianLangTech 2026: Transformer-Based Abusive Content Detection for Tamil Text Targeting Women on Social Media. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Kabilan Prasanna and Aryaman Arora. 2024. Irumozhi: Automatically classifying diglossia in tamil. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3096–3103.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, M Subramanian, and 1 others. 2023. Overview of shared-task on abusive comment detection in tamil and telugu. In *DravidianLangTech*.

Hanish Vigneshwar R, Nahul Alaguraj, Karthikeyan M, and Ratnavel Rajalakshmi. 2026. HNK@DravidianLangTech 2026: Investigating Grapheme-Level Normalization for Abusive Tamil Text Classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Saranya Rajiakodi, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, Ruba Priyadharshini, Kathiravan Pannerselvam, Rahul Ponnusamy, Bhuvaneswari Sivagnanam, Paul Buite-

laar, Jananayagan Jananayagan, Kishore Kumar Ponnusamy, and 1 others. 2025. Findings of the shared task on abusive tamil and malayalam text targeting women on social media: Dravidianlangtech@ naacl 2025. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 671–681.

Jacob Ratkiewicz, Michael Conover, Mark Meiss, Bruno Gonçalves, Alessandro Flammini, and Filippo Menczer. 2011. Detecting and tracking political abuse in social media. In *Proceedings of the International AAAI Conference on Web and social media*, volume 5, pages 297–304.

Oarisa Rebayet, Tahmima Hoque Eid, Fawzia Tabassum, and Hasan Murad. 2026. TriVector@DravidianLangTech 2026: Abusive Tamil Text Detection on Social Media Using Lexicon-Augmented Transformers. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Miftahul Jannat Rishta, Sumaiya Zaman, Shiti Chowdhury, and Hasan Murad. 2026. CUET_SYNTHETICA@DravidianLangTech 2026: Multi Architecture Transformer Ensemble for Detecting Abusive Tamil Text Targeting Women. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Mirudhula Sankar and Ratnavel Rajalakshmi. 2026. DLRG@DravidianLangTech 2026: Explainable Transformer-Based Detection of Abusive Tamil Text Targeting Women on Social Media. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Shrey Satapara, Prasenjit Majumder, Thomas Mandl, Sandip Modha, Hiren Madhu, Tharindu Ranasinghe, Marcos Zampieri, Kai North, and Damith Premasiri. 2023. Overview of the hasoc subtrack at fire 2022: Hate speech and offensive content identification in english and indo-aryan languages. In *Proceedings of the 14th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '22, page 4–7, New York, NY, USA. Association for Computing Machinery.

Shrey Satapara, Sarah Masud, Hiren Madhu, Md. Aflah Khan, Md. Shad Akhtar, Tanmoy Chakraborty, Sandip Modha, and Thomas Mandl. 2024. Overview of the hasoc subtracks at fire 2023: Detection of hate spans and conversational hate-speech. In *Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '23, page 10–12, New York, NY, USA. Association for Computing Machinery.

Sunil Saumya, Abhinav Kumar, and Jyoti Prakash Singh. 2021. Offensive language identification in dravidian code mixed social media text. In *Proceedings of the first workshop on speech and language technologies for Dravidian languages*, pages 36–45.

Nina Seemann, Yeong Su Lee, Julian Höllig, and Michaela Geierhos. 2023. The problem of varying annotations to identify abusive language in social media content. *Natural Language Engineering*, 29(6):1561–1585.

Deepawali Sharma, Vivek Kumar Singh, and Vedika Gupta. 2024. Tab hate: a target-based hate speech detection dataset in hindi. *Social Network Analysis and Mining*, 14(1):190.

Victoria L Stephenson, Brittany M Wickham, and Nicole M Capezza. 2018. Psychological abuse in the context of social media. *Violence and Gender*, 5(3):129–134.

Malliga Subramanian, Kogilavani Shanmugavadivel, Samyuktha K S, Santhiya D, Saranya P, Samritha A R, and Sivapriyan P. 2026. VISION@DravidianLangTech 2026: Abusive Tamil Text Detection Targeting Women on Social Media. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Nicolas Suzor, Molly Dragiewicz, Bridget Harris, Rosalie Gillett, Jean Burgess, and Tess Van Geelen. 2019. Human rights by design: The responsibilities of social media platforms to address gender-based violence online. *Policy & internet*, 11(1):84–103.

Rithikaa V, Sanjay Krishnan K, Nithya Varshini C N, and R S. Sumathi. 2026. PrimeLine@DravidianLangTech 2026: Abusive Tamil Comment Detection Using MuRIL. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Bertie Vidgen and Leon Derczynski. 2020. Directions in abusive language training data, a systematic review: Garbage in, garbage out. *Plos one*, 15(12):e0243300.

Bertie Vidgen, Alex Harris, Dong Nguyen, Rebekah Tromble, Scott A Hale, and Helen Margetts. 2019. Challenges and frontiers in abusive content detection. In *Proceedings of the third workshop on abusive language online*, pages 80–93.

Mohammad Zia Ur Rehman, Somya Mehta, Kuldeep Singh, Kunal Kaushik, and Nagendra Kumar. 2023. User-aware multilingual abusive content detection in social media. *Information Processing & Management*, 60(5):103450.

10 Appendix

10.1 Error Analysis

To better understand the failure modes of abusive Tamil text detection targeting women, we

conducted a qualitative error analysis on the predictions of the top five systems: nitc-hsr (A and Kumar, 2026), PRIMELINE_ABUSIVE, HNK (R et al., 2026), CUET_Synthetica (Rishta et al., 2026), and DPR (Prakash et al., 2026). Rather than estimating system-level performance, this analysis aims to identify repeated linguistic and contextual occurrences that contributed to misclassification. -media language patterns, target ambiguity, and annotation boundary conditions.

10.1.1 Setup

For error analysis, we examined the predictions from the top-5 systems (NIT-HSR, PRIMELINE_ABUSIVE, HNK (R et al., 2026), CUET_Synthetica (Rishta et al., 2026), DPR (Prakash et al., 2026)) and manually inspect a curated set of mispredicted instances from the first 250 samples of test set and we use it for qualitative analysis rather than estimating overall performance. We qualitatively analyze false positives (FP- gold Non-Abusive but predicted Abusive) and false negatives (FN- gold Abusive but predicted Non-Abusive) and grouping them into categories relevant to Tamil social-media abuse targeting women.

10.1.2 Common error categories

Masked or Obfuscated Abuse (False Negatives). Users often evade moderation by masking abusive terms using partial redaction or character substitution. Models that rely on surface lexical cues struggle when the key abusive word is hidden. In several such cases, systems including nitc-hsr (A and Kumar, 2026), HNK (R et al., 2026), and CUET_Synthetica (Rishta et al., 2026) predicted Non-Abusive despite the presence of masked offensive language. These errors suggest that current transformer models lack robustness to obfuscated abusive expressions and would benefit from normalization strategies capable of handling masked tokens.

Quoting, Reporting, and Counterspeech (False Positives). False positives also occur when users criticize or question abusive behavior using strong language. In such instances, the intent of the comment is corrective rather than abusive. Systems such as primeline_abusive_Tamil (V et al., 2026), HNK (R et al., 2026), and DPR (Prakash et al., 2026) occasionally predicted Abusive because the models reacted to emotionally expressive tokens without capturing the broader stance of the comment.

Profanity or Aggressive Tone Without Abusive Intent (False Positives). Several comments contain strong insults or aggressive tone but are labeled Non-Abusive under the task definition because they are not directed toward women. Systems including primeline_abusive_Tamil (V et al., 2026), HNK (R et al., 2026), CUET_Synthetica (Rishta et al., 2026), and DPR (Prakash et al., 2026) frequently predicted such instances as Abusive. This indicates that many models equate lexical signals of toxicity with abusive language, leading to overprediction of the abusive class.

Threat or Violent Expressions Without Clear Target (False Positives). Expressions containing violent or threatening verbs often trigger abusive predictions even when the target of the comment is unclear or when the statement is used rhetorically. In several examples, all top systems primeline_abusive_Tamil (V et al., 2026), HNK (R et al., 2026), CUET_Synthetica (Rishta et al., 2026), and DPR (Prakash et al., 2026) classified such comments as Abusive, despite the ground truth annotation being Non-Abusive. This suggests that transformer models strongly associate threat-related vocabulary with abusive intent regardless of contextual interpretation.

Code-Mixed and Noisy Social Media Language. Tamil social media comments frequently contain a mixture of Tamil and English words, transliterated expressions, and spelling variations. Although multilingual transformer models such as MuRIL and IndicBERT are designed to handle multilingual data, inconsistencies in spelling and informal writing styles still lead to errors. In several cases, CUET_Synthetica (Rishta et al., 2026) and nitc-hsr (A and Kumar, 2026) predicted Non-Abusive when abusive intent was present within code-mixed expressions.

Annotation Boundary Cases and Ambiguity. Finally, certain comments fall near the boundary of the annotation guidelines, where offensiveness depends on whether the insult is explicitly directed toward women or whether it represents general criticism. Such borderline cases produced inconsistent predictions across multiple systems including primeline_abusive_Tamil (V et al., 2026), HNK (R et al., 2026), and DPR (Prakash et al., 2026). These observations highlight how dataset definitions and annotation policies influence automated classification outcomes.

Overall, most misclassifications arise from re-

liance on surface lexical cues, limited ability to interpret discourse context, and difficulty handling noisy or obfuscated social media language.

Overview of the Shared Task on Multilevel Political Meme Classification in Tamil and Malayalam

Saranya Rajiakodi[1], Shunmuga Priya Muthusamy Chinnan[2], Premjith B[3], Subalalitha CN[4], Rahul Ponnusamy[2], Anshid K A[5], Bhuvaneswari Sivagnanam[1], Jananayagan V[1], Ragavan N[1], Santhini P[1], Bharathi Raja Chakravarthi[2]

[1]Central University of Tamil Nadu, India,
[2]Data Science Institute, University of Galway, Ireland
[3]Amrita Vishwa Vidhyapeetham, Tamil Nadu, India
[4]SRMIST, Tamil Nadu, India, [5]Mar Thoma College, Kerala, India

Abstract

This paper presents an overview of the Multi-Level Political Meme Classification shared task conducted at DravidianLangTech–ACL 2026. The task introduces a hierarchical two-level classification framework for Tamil and Malayalam political memes: Level 1 focuses on stance detection (Support/Praise vs. Troll/Oppose), while Level 2 identifies the political target (individual or party), conditioned on the predicted stance. The dataset was curated from social media platforms and manually annotated with strong inter-annotator agreement. A total of 64 teams registered for the shared task, of which 19 teams submitted official results using diverse multimodal approaches, including transformer-based text encoders, vision models, optical character recognition (OCR) pipelines, and hierarchical architectures. The results show that Level 1 stance detection achieved high macro-F1 scores across both Tamil and Malayalam, with several systems exceeding 0.96 F1, whereas Level 2 target identification remained substantially more challenging, especially for Malayalam. These findings highlight the importance of multimodal fusion, hierarchical reasoning, and robustness to OCR noise, cultural context, and class imbalance in political meme analysis.

Disclaimer: This paper (including figures, memes, and examples) contains political content and may include offensive, harmful, or derogatory language and imagery. All such material is reproduced solely for research, annotation, and educational purposes to study multilevel political meme classification. It does not represent the authors' views or endorsement of any ideology, party, or individual. Reader discretion is advised.

1 Introduction

People use social media platforms for communication and sharing their opinions on different domains including politics. Moreover, visual content, for example memes, play an important role in shaping public opinion. A meme consists of image and text to convey the attitude, humour and criticism on various topics, which have more impact than plain text (Trelles-Villanueva et al., 2025). In political contexts, memes were used as powerful tool for both supporting and ridicule political figures and parties (AlAfnan, 2025), (Anderau and Barbarrusa, 2024). This makes the automated analysis of political memes an important challenge in computational social science.

Significant progress has been made in meme analysis for high-resource languages such as English (Hossain et al., 2022), (Veeramani et al., 2024). Various image and text encoding techniques were employed for analysing memes including GPT-4 to generate contextual information by combining the image caption and the text in the meme, Bootstrapping Language-Image Pre-training (BLIP) to obtain the text describing the meme image with U-net Encapsulated Transformer (Ignacio et al., 2024), and Dual Contrastive Learning (DualCL) approach (Veeramani et al., 2024). However, low-resource Dravidian languages largely underexplored (Vivek et al., 2025). These languages present unique challenges such as the complex scripts (Chakravarthi et al., 2020), usage of culturally embedded political references to express sarcasm and irony (Ponnusamy et al., 2026). Generally, memes derive meaning from both visual and textual elements. A single image may convey different political interpretations depending on the supporting caption. Therefore, multimodal approaches, which can jointly reason over visual and textual content are required for addressing these challenges (Premjith et al., 2022), (Chakravarthi et al., 2024). In addition, the hierarchical reasoning, where the identification of stance is done first and determine the specific political target has not been systematically benchmarked for

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 101–113
July 3, 2026 ©2026 Association for Computational Linguistics

Tamil and Malayalam.

To address this gap, the Multi-Level Political Meme Classification shared task was organized the shared task on political meme classification at DravidianLangTech at ACL 2026. The task involves a hierarchical two-level classification, where the the first level objective is to classify whether a meme expresses Support/Praise or Troll/Oppose toward a political entity, and the second level objective is to determine whether the meme refers to an individual political figure or a political party/group, conditioned on the stance prediction. This paper presents the overview of the DravidianLangTech shared task on Multi-Level Political Meme Classification, hosted at ACL 2026.

A total of 19 teams participated in the shared task and submitted systems that used various vision models, text encoders, and fusion strategies. The participating teams used a variety of approaches including Late fusion of transformer-based text encoders (XLM-R, IndicBERT, MuRIL, mBERT) with vision encoders (Contrastive Language–Image Pretraining (CLIP), Residual Network (ResNet), ConvNeXt (Convolutional Next Network)), attention-based and gated cross-modal fusion, hierarchical and cascaded classification architectures, OCR-based text extraction pipelines, vision-language models fine-tuned with instruction prompts, and image-only systems leveraging face detection and political symbol matching.

The remainder of the paper is organized as follows. Sections 2 and 3 describe the shared task formulation and dataset construction process. Section 4 summarizes the methodologies adopted by the participating teams. Section 5 presents the experimental results and discussion. Finally, the conclusion, limitations, and future work are discussed in Sections 6, 7, 8.

2 Task Description

The goal of this task is to classify political memes at two levels using both the image and the text present in the meme. Participants are required to build systems that can identify the opinion expressed in a meme and the political entity it refers to.

Level 1 – Stance Classification. At this level, the system must decide whether a meme shows a positive or negative attitude toward a political person, party. This is a binary classification task with two labels: *Support/Praise* and *Troll/Oppose*.

Level 2 – Target Identification. After identifying the stance, the system must determine the specific target of the meme. The target can be either an individual political figure or a political party or group. This level provides more detailed information about who or what the meme is directed at and depends on the prediction made at Level 1.

Together, these two levels form a hierarchical classification structure, as shown in Table 1.

Level 1	Level 2
Support/Praise	Support for an individual person
	Support for a political party
Troll/Oppose	Troll against an individual person
	Troll against a political party

Table 1: Representative label combinations in the shared task.

This task involves several challenges. The meaning of a meme often comes from the combination of the image and the text, and using only one of them is usually not sufficient. In Tamil and Malayalam political memes (Fig 2), sarcasm, cultural context, and indirect expressions are common, which makes interpretation difficult. In addition, the dataset contains more trolling memes than supportive ones, leading to class imbalance. Finally, the same image can convey different meanings when paired with different text, so both image and text must be analyzed together. The shared task was hosted on the Codabench platform. Participants were provided with labelled training data and unlabelled test data for both Tamil and Malayalam. Systems were evaluated using macro-averaged F1 score across all label combinations.

3 Dataset Description

Language	Train	Test	Total
Tamil	802	201	1003
Malayalam	500	100	600

Table 2: Multimodal Meme Dataset Statistics

The dataset comprises two South Indian language corpora of political memes. The Tamil dataset consists of 1003 memes sourced from publicly available platforms, divided into 802 training instances and 201 test instances. Similarly, the Malayalam dataset consists of 600 political memes collected from publicly available sources, including Facebook, Instagram, and web searches via Google, divided into 500 training instances and

100 test instances to support supervised learning and evaluation. Fig 1 illustrates the steps involved in data curation, from collection through annotation to the final dataset.

3.1 Data Collection and Pre Processing

The dataset consists of political memes collected from Facebook, Instagram, and Google through automated web data extraction using the BeautifulSoup4 Python library, in combination with official platform APIs. Memes were identified through political hashtags and keywords. The collected images were cleaned before annotation, wherein duplicate images were removed and memes that were too unclear to interpret were discarded. To ensure privacy, user details, if any, were anonymized, and all images were converted to JPG format.

3.2 Data Annotation

Inter-annotator agreement (IAA) was measured to assess the consistency among annotators for both Tamil and Malayalam subsets. Two standard metrics were employed: Krippendorff's alpha and Fleiss' kappa.

Krippendorff's Alpha measures agreement among any number of annotators on any type of data, accounting for chance agreement as in Eq. (1):

$$\alpha = 1 - \frac{D_o}{D_e} \tag{1}$$

where D_o is the observed disagreement and D_e is the expected disagreement by chance.

Fleiss' Kappa measures agreement among a fixed number of annotators assigning items to categorical labels, corrected for chance as in Eq. (2):

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \tag{2}$$

where $\bar{P}$ is the mean observed agreement and $\bar{P}_e$ is the mean expected agreement by chance.

Both metrics were applied across two hierarchical annotation levels. For Tamil, agreement at Level 1 was high ($\alpha = 0.876$, $\kappa = 0.876$), indicating strong consistency in identifying the general stance of memes. At Level 2, agreement was moderate ($\alpha = 0.606$, $\kappa = 0.606$), reflecting the greater difficulty in agreeing on more fine-grained political categories.

For Malayalam, Level 1 agreement was similarly strong ($\alpha = 0.817$), showing that annotators were largely consistent in their general assessments. Level 2 achieved perfect agreement ($\alpha = 1.000$), meaning annotators were in complete agreement when identifying specific political intent.

Language	Level	Krippendorff's α	Fleiss' κ
Tamil	Level 1	0.876	0.876
	Level 2	0.606	0.606
Malayalam	Level 1	0.817	–
	Level 2	1.000	–

Table 3: Inter-annotator agreement scores for Tamil and Malayalam.

Overall, both languages show reliable annotation quality, with particularly strong agreement at the broader classification level across both subsets.

3.2.1 Annotator Demographics

The annotation was carried out by two separate teams for Tamil and Malayalam. For Tamil, three postgraduate students aged between 20 and 25, comprising both male and female annotators, completed the annotation. For Malayalam, the annotation was divided across two levels: Level 1 was handled by four annotators (two male, two female) aged 21 to 27, while Level 2 was completed by two male annotators aged 27 to 37. All annotators were proficient in English along with their respective languages. Table 4 provides the combined demographic details.

Annotation guidelines. Annotators were instructed to jointly interpret both the textual and visual components of each meme before assigning labels. For Level 1 (stance classification), memes were categorized as *Support/Praise* or *Troll/Oppose* based on the overall political stance expressed toward a political entity such as a politician, party, or group. For Level 2 (target identification), annotators identified the primary political target of the meme, including political leaders, political parties, organizations, or communities.Annotators were additionally instructed to consider sarcasm, symbolic imagery, implicit political messaging, and culturally grounded references while labeling memes. Ambiguous or borderline cases were resolved through discussion with an additional annotator to improve annotation consistency.

4 Participant Methodology

A total of 19 teams submitted systems for the shared task covering Tamil and Malayalam political meme classification. Most participants built

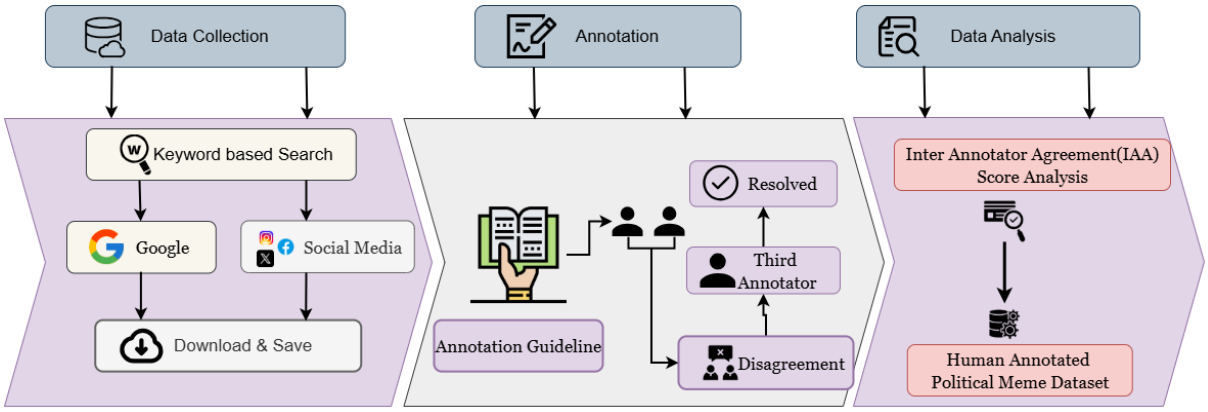


Figure 1: Steps Involved in Data Curation.

(a) Level 1: Support/Praise
Level 2: Support for Individual Person (Malayalam)

(b) Level 1: Troll/Oppose
Level 2: Troll against Individual Person (Malayalam)

(c) Level 1: Support/Praise
Level 2: Support for Party (Tamil)

(d) Level 1: Troll/Oppose
Level 2: Troll against Individual Person (Tamil)

Figure 2: Example memes with multi-level classification labels.

Language	Level	Gender	Age	Qualification
Tamil	1, 2	Female	20–25	PG
		Female	20–25	PG
		Male	20–25	PG
Malayalam	1	Female	21–27	UG
		Female	21–27	UG
		Male	21–27	UG
		Male	21–27	PG
	2	Male	27–37	PG
		Male	27–37	PhD

Table 4: Demographic profile of the annotation teams for Tamil and Malayalam (UG: Undergraduate; PG: Postgraduate).

systems that process both the text and image content of memes, which reflects the nature of the task where meaning often arises from the combination of visual and textual elements. Table 5 shows a high level overview of participant approaches. This section summarises the main approaches taken by participants.

4.1 Combining Text and Image Features

Most systems processed text and image content separately and combined the resulting representations before classification. The most common approach was to concatenate the text and image feature vectors and pass the combined representation through classification layers. Codecrafters, CUET-2567 (Dutta et al., 2026), CUET-DoubleA, Memeocracy, SJK, SSN-CSE-DiNa, susmitha, nitc-hsr, and RMS all followed this design due to its simplicity and consistent performance.

Some teams went beyond simple concatenation. CUETEuropa used an attention mechanism where the text representation guided which parts of the image representation to focus on, followed by a gating step that balances the contribution of each modality. Meme_Alchemists fed the stance prediction from Level 1 as an additional input to the Level 2 classifier so that the target prediction is informed by the predicted stance. MindBridge applied label constraints during training and inference to prevent the model from producing outputs that are inconsistent across the two levels. HAG Signals, IndiLangTech, Semantica, and Synapse (K P et al., 2026) used a single pre-trained vision-language model to process both text and image together without a separate combination step.

4.2 Text Representations

XLM-RoBERTa was the most frequently used model for encoding meme text. Codecrafters, Memeocracy, SSN-CSE-DiNa, susmitha, and CUETEuropa (for Malayalam) used it for its coverage of multiple languages including Tamil and Malayalam. IndicBERT, which was trained on a large collection of Indian language text, was used by CUET-DoubleA, CUETEuropa (for Tamil), SJK, and Team_One. MuRIL was used by Meme_Alchemists, and mBERT was used by CUET-2567 (Dutta et al., 2026) and CUET-DoubleA.

Semantica fine-tuned a seven-billion-parameter vision-language model using a low-rank adaptation technique with instruction-style prompts for both languages. Synapse (K P et al., 2026) applied a similar adaptation method to a two-billion-parameter model and deliberately gave more weight to the text content than the image during training, as Tamil political memes often carry their primary meaning through language. IndiLangTech built a two-stage system using a four-billion-parameter instruction-tuned model, where the first stage predicts the stance of a meme and its output is passed to the second stage to identify the target, allowing each stage to focus on a well-defined sub-task.

4.3 Image Representations

CLIP was the most widely used model for extracting image features. Codecrafters, MindBridge, nitc-hsr, Meme_Alchemists, susmitha, and Cyberpunk (Abir, 2026) used CLIP either alone or alongside other encoders. Memeocracy used ConvNeXt-Tiny, Team_One and SSN-CSE-DiNa used ResNet-18, SJK used VGG16, DLRG used EfficientNet-B3, and CUETEuropa combined CLIP, ResNet, and EfficientNet.

Cyberpunk (Abir, 2026) designed a system that does not use any text extraction and relies entirely on image content. It used CLIP image embeddings combined with features derived from detected faces and the visual similarity of party logos and symbols collected from public sources. This approach is not affected by the quality of text in the image, which can be unreliable in political memes that use stylised or low-resolution fonts.

4.4 Text Extraction from Meme Images

CUET-2567, CUET-DoubleA, CUETEuropa, Meme_Alchemists, MindBridge, Memeocracy,

Team	Lang.		Text Model	Vision Model	Fusion Strategy	PT	IH	OCR	MT	HC
	Ta	Ml								
Codecrafters	✓	✓	XLM-RoBERTa	CLIP / ViT	Multi-modal late fusion	✓	✗	✗	✗	✗
CUET-2567 (Dutta et al., 2026)	✓	✓	mBERT	ViT (patch16-224)	Late fusion (concat.)	✓	✗	✓	✗	✗
CUET-DoubleA	✓	✓	mBERT, IndicBERT, RoBERTa	ViT, ResNet, CLIP	Late fusion (concat.)	✓	✓	✓	✗	✓
CUETEuropa	✓	✓	IndicBERT, Qwen / XLM-R	CLIP, Qwen VL / ResNet, EfficientNet	Cross-attn. + gating	✓	✓	✓	✓	✓
Cyberpunk	✓	✓	— (OCR-free)	OpenCLIP	Image-only (CLIP+face/logo)	✓	✓	✗	✗	✓
DLRG	✓	✗	—	EfficientNet-B3	Image-only (CNN)	✓	✗	✗	✗	✗
HAG Signals	✓	✓	VLM (zero-shot)	VLM (zero-shot)	Zero-shot VLM	✓	✗	✗	✗	✓
IndiLangTech	✓	✓	Gemma 3 4B (Agent-1,2)	Gemma 3 4B (multimodal)	Agentic hierarchical (instruct.)	✓	✗	✗	✗	✓
Meme_Alchemists	✓	✓	MuRIL	CLIP ViT-B/32	Late fusion + L1→L2 cond.	✓	✓	✓	✓	✓
Memeocracy	✓	✓	XLM-RoBERTa	ConvNeXt-Tiny	Late fusion (concat.)	✓	✓	✓	✓	✗
MindBridge	✓	✓	CLIP (text enc.)	CLIP (vision enc.)	CLIP + hierarchical masking	✓	✓	✓	✓	✓
nitc-hsr	✓	✓	CLIP (text enc.)	CLIP (vision enc.)	Multimodal CLIP fusion	✓	✗	✗	✗	✗
RMS	✓	✓	Transformer enc.	Pretrained vision backbone	Concat. + gated fusion	✓	✓	✓	✓	✗
Semantica	✓	✓	Qwen2.5-VL-7B (QLoRA)	Qwen2.5-VL-7B (QLoRA)	VLM fine-tuning (instruct.)	✓	✓	✗	✗	✓
SJK	✓	✓	IndicBERT	VGG16	Late fusion (concat.)	✓	✗	✓	✓	✗
SSN-CSE-DiNa	✓	✓	XLM-RoBERTa	ResNet-18	Late fusion + MLP	✓	✗	✓	✗	✗
susmitha	✓	✓	XLM-RoBERTa	CLIP ViT-B/32	Late fusion (concat.)	✓	✗	✓	✓	✗
Synapse (K P et al., 2026)	✓	✗	Qwen3-VL-2B (LoRA)	Qwen3-VL-2B (LoRA)	VLM text-primacy PEFT (LoRA)	✓	✗	✗	✗	✗
Team_One	✓	✓	IndicBERT / XLM-R	None / ResNet-18	Text-only (Ta); Late fusion (Ml)	✓	✓	✗	✗	✗

Legend: Ta=Tamil, Ml=Malayalam, PT=Pre-trained Models, IH=Imbalance Handling, OCR=OCR Used, MT=Multi-task Learning, HC=Hierarchical Classification

Table 5: Overview of Participants System

RMS, SJK, SSN-CSE-DiNa, and susmitha extracted visible text from meme images using optical character recognition tools, including EasyOCR, PaddleOCR, and Tesseract. Several of these teams noted that text extraction introduced noise into their pipeline, particularly for Tamil script, because memes often use non-standard fonts and image compression that reduces text clarity. Some teams applied text cleaning steps after extraction to reduce this noise before encoding.

4.5 Handling Class Imbalance

The datasets for both languages contained a much larger number of troll-class examples compared to support-class examples, with the imbalance being particularly severe in the Malayalam dataset. Teams addressed this in different ways. Team_One and Semantica increased the number of training examples for the minority class by repeating samples. Team_One additionally lowered the decision threshold at inference time to produce more support-class predictions. Memeocracy, MindBridge, CUETEuropa, and RMS assigned higher penalties to errors on the minority class during training using weighted loss functions. Semantica also addressed the tendency of the model to misclassify sarcastic troll memes as support by adding explicit instructions in the prompt describing how sarcasm and irony should be interpreted. CUET-DoubleA generated additional Tamil training examples by translating existing samples through other Indian and high-resource languages and back into Tamil.

4.6 Two-Level Classification

Several teams specifically addressed the relationship between the two classification levels in their system design. Level 1 requires the system to predict whether a meme expresses support or opposition, while Level 2 requires identifying whether the target is an individual or a political party. IndiLangTech separated the two levels into distinct fine-tuned stages, with the output of the first stage passed as context to the second. Meme_Alchemists trained a shared model with two output heads and fed Level 1 predictions directly into the Level 2 head. MindBridge applied masking to prevent logically contradictory label combinations. Cyberpunk (Abir, 2026) used a two-stage prediction process specifically for the Malayalam target category to handle the rare intersection class. CUET-DoubleA applied a logical mapping after prediction to ensure consistency between the two levels.

4.7 Language Coverage

Seventeen of the 19 teams submitted results for both Tamil and Malayalam. DLRG and Synapse (K P et al., 2026) participated only in the Tamil subtask. Most teams trained separate models for each language to account for differences in dataset size and class distribution, while keeping the overall system design the same across both languages.

5 Results and Discussion

Table 6 and Table 7 present the official results for the Malayalam and Tamil sub-tasks respectively. Systems are ranked by the average macro-F1 score across both levels. Results of all runs of all systems

are in Appendix 10, 11

The experimental results across both Tamil and Malayalam sub-tasks reveal clear trends in system performance, model design choices, and the challenges inherent to multi-level political meme classification. Across all submissions, performance on Level 1, which focuses on stance detection, is consistently higher than on Level 2, which requires identifying the political target. This difference reflects the increased complexity of target identification, where cues are often implicit, culturally grounded, or conveyed through sarcasm and visual symbolism rather than explicit textual mentions.

Systems generally achieve macro-F1 scores above 0.90 for Level 1 in both languages, indicating that current multimodal representations are effective at capturing overall sentiment or stance. In contrast, Level 2 performance is notably lower, particularly for Malayalam. This gap can be attributed to several factors, including stronger class imbalance, subtler visual metaphors, and a heavier reliance on contextual political knowledge in Malayalam memes. Tamil systems tend to perform better on Level 2, likely due to a larger dataset size and more explicit linguistic or visual markers that signal the intended target.

The strongest systems demonstrate that architectural design and task decomposition play a critical role in overall performance. Cyberpunk, which ranks highest for Malayalam, achieves competitive results without relying on OCR or textual encoders, instead focusing entirely on visual information such as faces and political symbols. This suggests that for languages where OCR quality is unreliable due to stylised fonts or low-resolution text, image-centric approaches can offer a robust alternative. In contrast, the best-performing Tamil systems, particularly IndiLangTech (Kumar et al., 2026) and CUETEuropa, rely heavily on textual and cross-modal interactions, highlighting the language-specific nature of effective feature extraction strategies.

Several teams (Appendix Table 7, 9) explicitly modeled the hierarchical relationship between Level 1 and Level 2 labels, either by cascading predictions or by constraining the output space. These approaches generally lead to more consistent predictions across levels and reduce logically invalid label combinations. However, improvements in consistency do not always translate into large gains in macro-F1, especially when errors at Level 1 propagate to Level 2 in cascaded systems. This trade-off indicates that while hierarchical modeling is beneficial, it must be carefully designed to avoid amplifying early-stage errors.

A common source of error across many systems arises from the use of OCR for text extraction. Although OCR enables access to valuable linguistic information, especially in Tamil, it often introduces noise due to non-standard fonts, image compression, and creative typography common in memes. Several teams reported that noisy OCR output adversely affected downstream text encoders unless extensive cleaning was applied. This observation explains why some OCR-free or image-dominant systems remain competitive despite ignoring textual content altogether.

Class imbalance presents another major challenge, particularly for Level 2 categories in Malayalam, where troll-related classes dominate the dataset. While weighted loss functions, oversampling, and threshold adjustment help mitigate this issue, many systems still exhibit reduced recall for minority classes. This suggests that data-level solutions alone are insufficient and that more targeted modeling strategies are needed to handle rare but semantically important categories.

An interesting pattern emerges when comparing fusion strategies. Simple late fusion through feature concatenation remains a strong baseline and performs competitively against more complex attention-based architectures. This indicates that representation quality and task-specific training often outweigh the benefits of sophisticated fusion mechanisms, especially under limited data conditions. At the same time, systems using attention or instruction-tuned vision–language models show advantages in handling nuanced cases, particularly for Level 2 target identification in Tamil. Eventually, the results highlight that effective political meme classification requires balancing multimodal integration, hierarchical reasoning, and robustness to noisy inputs. While current systems demonstrate strong performance for stance detection, the persistent difficulty of target identification underscores the need for future research on culturally informed representations, sarcasm-aware modeling, and improved annotation of implicit political references in low-resource languages.

6 Conclusion

This shared task introduced a benchmark for hierarchical political meme classification in low-resource Dravidian languages, namely Tamil and

Team	Level 1				Level 2				Avg-F1	Rank
	ACC	P	R	F1	ACC	P	R	F1		
Cyberpunk (Abir, 2026)	0.9600	0.9698	0.9600	0.9638	0.6300	0.6367	0.6300	0.6222	**0.7930**	1
CUET-2567	0.9600	0.9216	0.9600	0.9404	0.6300	0.6124	0.5900	0.5148	0.7276	2
Semantica	0.8400	0.9680	0.8400	0.8861	0.5300	0.6253	0.5300	0.5494	0.7178	3
IndiLangTech (Kumar et al., 2026)	0.9400	0.9208	0.9400	0.9303	0.4900	0.3933	0.4900	0.4302	0.6802	4
Synapse	0.9200	0.9200	0.9200	0.9200	0.5000	0.4442	0.5000	0.4256	0.6728	5
Meme_Alchemists	0.9600	0.9216	0.9600	0.9404	0.5000	0.4675	0.5000	0.4048	0.6726	6
MindBridge	0.9600	0.9216	0.9600	0.9404	0.4700	0.3592	0.4700	0.4006	0.6705	7
DLRG	0.9200	0.9200	0.9200	0.9404	0.4500	0.3921	0.4500	0.4095	0.6647	8
CUETEuropa_Rathijit	0.9600	0.9216	0.9600	0.9404	0.4900	0.3740	0.4900	0.3819	0.6612	9
CUETEuropa_Joy	0.9600	0.9216	0.9600	0.9404	0.4900	0.3740	0.4900	0.3819	0.6612	9
SJK	0.9600	0.9216	0.9600	0.9404	0.4600	0.3439	0.4600	0.3742	0.6573	10
Memeocracy	0.9600	0.9506	0.9600	0.9535	0.3800	0.4470	0.3800	0.3545	0.6540	11
Codecrafters	0.9600	0.9216	0.9600	0.9404	0.3500	0.3055	0.3500	0.3115	0.6260	12
Team_One	0.8700	0.9313	0.8700	0.8979	–	–	–	–	0.4490	13

Table 6: Results for the Malayalam sub-task. Systems ranked by average macro-F1 across Level 1 and Level 2. ACC=Accuracy, P=Precision, R=Recall. Team_One did not submit Level 2 predictions for Malayalam.

Team	Level 1				Level 2				Avg-F1	Rank
	ACC	P	R	F1	ACC	P	R	F1		
IndiLangTech (Kumar et al., 2026)	0.9204	0.9147	0.9204	0.9159	0.6915	0.6930	0.6915	0.6898	**0.8029**	1
CUETEuropa_Rathijit	0.9204	0.9161	0.9204	0.9175	0.6716	0.6571	0.6716	0.6619	0.7897	2
CUETEuropa_Joy	0.9204	0.9161	0.9204	0.9175	0.6716	0.6571	0.6716	0.6619	0.7897	2
DLRG	0.9055	0.8986	0.9055	0.8883	0.7015	0.6740	0.7015	0.6842	0.7862	3
Semantica	0.8706	0.8979	0.8706	0.8805	0.6667	0.6936	0.6667	0.6785	0.7795	4
CUET-2567	0.9005	0.8921	0.9005	0.8806	0.6965	0.6652	0.6965	0.6556	0.7681	5
Cyberpunk (Abir, 2026)	0.9254	0.9295	0.9254	0.9271	0.5771	0.6558	0.5771	0.6060	0.7666	6
MindBridge	0.8856	0.8802	0.8856	0.8826	0.6965	0.6408	0.6965	0.6471	0.7648	7
SJK	0.8955	0.8822	0.8955	0.8765	0.6766	0.6521	0.6766	0.6487	0.7626	8
CUET-DoubleA	0.9055	0.9147	0.9055	0.8807	0.6617	0.6883	0.6617	0.6403	0.7605	9
Memeocracy	0.9055	0.8957	0.9055	0.8968	0.6020	0.6582	0.6020	0.6202	0.7585	10
RMS	0.8756	0.8554	0.8756	0.8608	0.6318	0.6151	0.6318	0.6156	0.7382	11
Meme_Alchemists	0.8806	0.8660	0.8806	0.8403	0.6766	0.5977	0.6766	0.5905	0.7154	12
Susmitha	0.8706	0.7580	0.8706	0.8104	0.6816	0.4646	0.6816	0.5525	0.6814	13
Synapse	0.7512	0.8414	0.7512	0.7840	0.4428	0.6117	0.4428	0.4885	0.6362	14
Codecrafters	0.8706	0.7580	0.8706	0.8104	0.1841	0.0339	0.1841	0.0572	0.4338	15

Table 7: Results for the Tamil sub-task. Systems ranked by average macro-F1 across Level 1 and Level 2. ACC=Accuracy, P=Precision, R=Recall.

Malayalam. Political memes present a challenging multimodal problem as they combine textual content, visual cues, and socio-political context to convey stance, criticism, sarcasm, and ideological positioning. To address this challenge, we proposed a two-level hierarchical framework where systems first predict the stance of the meme (Support/Praise vs. Troll/Oppose) and subsequently identify the political target (Individual vs. Party/Group). A total of 19 teams participated in the shared task and explored diverse modeling strategies, including multilingual transformer-based text encoders, visual backbone networks, OCR-enhanced pipelines, cross-modal fusion architectures, and vision–language models. The best-performing systems achieved macro F1-scores of 0.8029 for Tamil and 0.7930 for Malayalam, demonstrating the potential of multimodal approaches for political meme understanding in low-resource languages. The results highlight the importance of combining textual and visual signals for effective meme interpretation. However, performance variations across systems indicate that multimodal political meme analysis remains a challenging problem. We hope that this shared task dataset, benchmark, and system analyses will facilitate future research on multimodal political discourse, sarcasm detection, and stance analysis in low-resource languages.

7 Limitations

This section discusses the limitations of this work. Here, the dataset was collected from social media platforms, which may introduce sampling bias and may not fully represent the diversity of political discourse across different demographic groups and

regions. Social media data often reflects platform-specific communication styles and meme cultures. Variations in spelling, grammar, and transliteration may introduce additional challenges for automated systems. Another limitation is that, political memes often rely on subtle multimodal cues such as sarcasm, satire, symbolic imagery, and cultural references. Models may struggle to capture these nuanced signals, particularly when the meaning emerges from the interaction between text and visual content. The relatively limited size of the dataset and the scarcity of large-scale pretrained multimodal resources for Tamil and Malayalam is another limitation of this work.

8 Future Work

Future work may explore larger multilingual political meme datasets covering additional Indic languages and political contexts. Cross-lingual transfer learning between Tamil and Malayalam memes using multilingual vision-language models represents another promising direction. Further improvements may also be achieved through OCR-aware multimodal architectures, sarcasm-aware reasoning mechanisms, and explainable multimodal systems capable of identifying fine-grained political symbolism and contextual relationships. Future work could also explore larger datasets, richer multimodal representations, and improved annotation strategies for capturing complex political discourse.

9 Ethical Statement

This shared task involves political memes that may contain satire, trolling, or potentially offensive language. The data set was collected from publicly accessible sources and user identifiers were anonymized to protect privacy.Although strong IAA was achieved, the interpretation of political memes can be subjective. Models developed for political meme classification should be used responsibly. This benchmark data set is intended solely for academic research in multimodal NLP. Researchers are encouraged to ensure transparency, fairness, and respect for democratic principles when deploying related technologies.

Usage of AI Declaration

We used AI-based tools to support language editing and formatting of this manuscript (e.g., improving clarity, grammar, and LaTeX presentation). All technical content, experimental results, interpretations, and final wording were reviewed and verified by the authors. No AI system was used to generate or alter the dataset, annotations, or official leaderboard results.

Acknowledgements

This work was conducted with the financial support from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289_P2(Insight_2), supported in part of Science Foundation Ireland Centre for Research Training in Artificial Intelligence under Grant No. 18/CRT/6223.

References

Shahad Abir. 2026. CYBERPUNK@DravidianLangTech 2026: Multimodal Political Meme Classification using CLIP and Logo Similarity. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Mohammad Awad AlAfnan. 2025. The role of memes in shaping political discourse on social media. *Studies in Media and Communication*, 13(2):1–10.

Glenn Anderau and Daniel Barbarrusa. 2024. The function of memes in political discourse. *Topoi*, 43(5):1529–1546.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Shardul Suryawanshi, Navya Jose, Elizabeth Sherly, and John P McCrae. 2020. 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*, pages 21–24.

Bharathi Raja Chakravarthi, Saranya Rajiakodi, Rahul Ponnusamy, Kathiravan Pannerselvam, Ramachandran Rajalakshmi, Hariharan LekshmiAmmal, Anshid Kizhakkeparambil, Susminu S Kumar, Bhuvaneswari Sivagnanam, Charmathi Rajkumar, and 1 others. 2024. Overview of shared task on multitask meme classification - unraveling misogynistic and trolls in online memes. In *Proceedings of the Fourth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 139–144.

Arka Dutta, Anindya Majumder, Adnan Faisal, and Hasan Murad. 2026. CUET-2567@DravidianLangTech-ACL 2026: Multimodal Stance and Target Identification in Dravidian Political Memes. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Eftekhar Hossain, Omar Sharif, and Mohammed Moshiul Hoque. 2022. Memosen: A multimodal dataset for sentiment analysis of memes. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1542–1554.

Md. Ajwad Hossain. 2026. RMS@DravidianLangTech 2026: Multimodal Gated Fusion for Hierarchical Tamil Political Meme Classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Marvin John Ignacio, Thanh Tin Nguyen, Hulin Jin, and Yong-Guk Kim. 2024. Meme analysis using llm-based contextual information and u-net encapsulated transformer. *IEEE Access*, 12:125993–126005.

Suriya K P, Durai Singh K, Gnanasabesan G, Ganesh Sundhar, Hari Krishnan N, and Jyothish Lal G. 2026. SYNAPSE@DravidianLangTech 2026: Multi-Level Political Meme Classification for Tamil and Malayalam. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Saurabh Kumar, Vivekananda G, Sanasam Ranbir Singh, and Sukumar Nandi. 2026. IndiLangTech@DravidianLangTech 2026: Hierarchical Modeling for Multi-Level Political Meme Classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Rahul Ponnusamy, Saranya Rajiakodi, Bhuvaneswari Sivagnanam, Anshid Kizhakkeparambil, Dhruv Sharma, Paul Buitelaar, and Bharathi Raja Chakravarthi. 2026. Fuse-md: A culturally-aware multimodal model for detecting misogyny memes. *Natural Language Processing Journal*, page 100197.

B Premjith, Bharathi Raja Chakravarthi, Malliga Subramanian, B Bharathi, Soman Kp, V Dhanalakshmi, K Sreelakshmi, Arunaggiri Pandian, and Prasanna Kumaresan. 2022. Findings of the shared task on multimodal sentiment analysis and troll meme classification in dravidian languages. In *Proceedings of the second workshop on speech and language technologies for Dravidian languages*, pages 254–260.

Alicia Trelles-Villanueva, Rafael Carrasco-Polaino, and Miguel Ángel Sánchez de la Nieta. 2025. Humor, persuasion, and politics: The impact of memes on x during the 2024 u.s. presidential election campaign. *VISUAL REVIEW. International Visual Culture Review / Revista Internacional de Cultura Visual*, 17(4):313–324.

Junain Uddin, Rahul Datta, Taha Ibne Abdullah, and Hasan Murad. 2026. Semantica@DravidianLangTech 2026: Vision-Language Models for Hierarchical Political Meme Classification in Tamil and Malayalam. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Hariram Veeramani, Surendrabikram Thapa, Mohammad Salman, Mohammad Ali Moni, and Usman Naseem. 2024. Umml: Layout-aware text-image fusion for unified multilingual multimodal learning. In *2024 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 174–182. IEEE.

Karnati Vivek, Sai Mani Garrepalli, PVS Ajay Krishna, Inti Charan Kumar, Ravi Teja Gedela, and Sasibhushara Rao Pappu. 2025. Telmemood: A multimodal dataset for sentiment analysis of telugu memes. In *2025 3rd International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC)*, pages 893–899. IEEE.

A Appendix

Table 8: Tamil: Classification examples with sample images.

Image	Actual (Level 1 → Level 2)	Team	Pred Level 1	Pred Level 2
	Troll/Oppose → Against Party	INDILANGTECH (Kumar et al., 2026) ✓	Troll/Oppose	Against Party
		CUETEUROPA_RATHIJIT ✗	Troll/Oppose	Against Person
		CUETEUROPA_JOY ✗	Troll/Oppose	Against Person
		DLRG ✗	Troll/Oppose	Against Person
		SEMANTICA (Uddin et al., 2026) ✗	Troll/Oppose	Against Person
		CUET-2567 ✓	Troll/Oppose	Against Party
		Cyberpunk (Abir, 2026) ✓	Troll/Oppose	Against Party
		MINDBRIDGE ✓	Troll/Oppose	Against Person
		SJK ✓	Troll/Oppose	Against Party
		CUET-DOUBLEA ✓	Troll/Oppose	Against Party
		MEMOCRACY✓	Troll/Oppose	Against Party
	Troll/Oppose → Against Person	INDILANGTECH (Kumar et al., 2026)✗	Troll/Oppose	Against Party
		CUETEUROPA_RATHIJIT ✓	Troll/Oppose	Against Person
		CUETEUROPA_JOY ✓	Troll/Oppose	Against Person
		DLRG ✓	Troll/Oppose	Against Person
		SEMANTICA ✓	Troll/Oppose	Against Person
		CUET-2567 ✓	Troll/Oppose	Against Person
		Cyberpunk (Abir, 2026) ✓	Troll/Oppose	Against Party
		MINDBRIDGE ✓	Troll/Oppose	Against Person
		SJK ✓	Troll/Oppose	Against Person
		CUET-DOUBLEA ✓	Troll/Oppose	Against Person
		MEMOCRACY✓	Troll/Oppose	Against Person

Table 9: Malayalam: Classification examples with sample images from top 6 teams.

Image	Actual (Level 1 → Level 2)	Team	Pred Level 1	Pred Level 2
	Support/Praise → Support for Person	Cyberpunk (Abir, 2026) ✗	Troll/Oppose	Against Person
		CUET-2567 ✗	Troll/Oppose	Against Person
		SEMANTICA ✓	Support/Praise	Support for Person
		INDILANGTECH (Kumar et al., 2026) ✗	Troll/Oppose	Against Person
		SYNAPSE ✗	Troll/Oppose	Against Person
		ALCHEMIST ✗	Troll/Oppose	Against Person
		MINDBRIDGE ✗	Troll/Oppose	Against Person
		DLRG ✗	Troll/Oppose	Against Person
		CUETEUROPA RATHIJIT✗	Troll/Oppose	Against Person
		CUETEUROPA JOY✗	Troll/Oppose	Against Person
		SJK✗	Troll/Oppose	Against Person
	Troll/Oppose → Against Person	Cyberpunk (Abir, 2026) ✓	Troll/Oppose	Against Person
		CUET-2567 ✓	Troll/Oppose	Against Person
		SEMANTICA ✗	Support/Praise	Support for Party
		INDILANGTECH (Kumar et al., 2026) ✗	Support/Praise	Support for Person
		SYNAPSE ✓	Troll/Oppose	Against Person
		ALCHEMIST ✓	Troll/Oppose	Against Person
		MINDBRIDGE ✗	Troll/Oppose	Against Person
		DLRG ✗	Troll/Oppose	Against Person
		CUETEUROPA RATHIJIT✗	Troll/Oppose	Against Party
		CUETEUROPA JOY✗	Troll/Oppose	Against Party
		SJK✗	Troll/Oppose	Against Person

Rank	Team	Run	Avg F1	L1 F1	L2 F1	L1 Acc	L1 Prec	L1 Rec	L2 Acc	L2 Prec	L2 Rec
1	IndiLangTech (Kumar et al., 2026)	Run3	0.8029	0.9159	0.6898	0.9204	0.9147	0.9204	0.6915	0.6930	0.6915
2	CUETEuropa_RATHIJIT	Run2	0.8027	0.9175	0.6879	0.9204	0.9161	0.9204	0.7363	0.6985	0.7363
2	CUETEuropa - JOY	Run2	0.8027	0.9175	0.6879	0.9204	0.9161	0.9204	0.7363	0.6985	0.7363
3	IndiLangTech (Kumar et al., 2026)	Run1	0.8022	0.9159	0.6885	0.9204	0.9147	0.9204	0.7065	0.6754	0.7065
4	CUETEuropa_RATHIJIT	Run3	0.7897	0.9175	0.6619	0.9204	0.9161	0.9204	0.6716	0.6571	0.6716
4	CUETEuropa - JOY	Run3	0.7897	0.9175	0.6619	0.9204	0.9161	0.9204	0.6716	0.6571	0.6716
5	DLRG	Single	0.7862	0.8883	0.6842	0.9055	0.8986	0.9055	0.7015	0.6740	0.7015
6	Semantica	Run1	0.7795	0.8805	0.6785	0.8706	0.8979	0.8706	0.6667	0.6936	0.6667
6	Semantica	Run2	0.7795	0.8805	0.6785	0.8706	0.8979	0.8706	0.6667	0.6936	0.6667
7	CUET-DoubleA	Run2	0.7744	0.8807	0.6680	0.9055	0.9147	0.9055	0.7164	0.7066	0.7164
8	IndiLangTech (Kumar et al., 2026)	Run2	0.7733	0.9033	0.6432	0.9104	0.9022	0.9104	0.6965	0.6363	0.6965
9	CUETEuropa - JOY	Run1	0.7714	0.9028	0.6400	0.9154	0.9106	0.9154	0.6617	0.6510	0.6617
9	CUETEuropa_RATHIJIT	Run1	0.7714	0.9028	0.6400	0.9154	0.9106	0.9154	0.6617	0.6510	0.6617
10	CUET-2567	Single	0.7681	0.8806	0.6556	0.9005	0.8921	0.9005	0.6965	0.6652	0.6965
11	RMS (Hossain, 2026)	Run1	0.7672	0.8763	0.6581	0.8806	0.8731	0.8806	0.6617	0.6580	0.6617
12	Cyberpunk (Abir, 2026)	Run2	0.7666	0.9271	0.6060	0.9254	0.9295	0.9254	0.5771	0.6558	0.5771
13	MindBridge	Single	0.7648	0.8826	0.6471	0.8856	0.8802	0.8856	0.6965	0.6408	0.6965
14	Cyberpunk (Abir, 2026)	Run1	0.7629	0.9007	0.6252	0.8955	0.9088	0.8955	0.6020	0.6667	0.6020
15	SJK	Single	0.7626	0.8765	0.6487	0.8955	0.8822	0.8955	0.6766	0.6521	0.6766
16	CUET-DoubleA	Run3	0.7605	0.8807	0.6403	0.9055	0.9147	0.9055	0.6617	0.6883	0.6617
17	Memocracy	Single	0.7585	0.8968	0.6202	0.9055	0.8957	0.9055	0.6020	0.6582	0.6020
18	CUET-DoubleA	Run1	0.7571	0.8686	0.6455	0.8905	0.8746	0.8905	0.6866	0.6410	0.6866
19	RMS (Hossain, 2026)	Run2	0.7525	0.8884	0.6165	0.8955	0.8854	0.8955	0.6418	0.6059	0.6418
20	RMS (Hossain, 2026)	Run3	0.7382	0.8608	0.6156	0.8756	0.8554	0.8756	0.6318	0.6151	0.6318
21	Alchemists	Single	0.7154	0.8403	0.5905	0.8806	0.8660	0.8806	0.6766	0.5977	0.6766
22	SUSMITHA	Single	0.6815	0.8104	0.5525	0.8706	0.7580	0.8706	0.6816	0.4646	0.6816
23	Synapse (K P et al., 2026)	Single	0.6362	0.7840	0.4885	0.7512	0.8414	0.7512	0.4428	0.6117	0.4428
24	Codecrafters	Single	0.4338	0.8104	0.0572	0.8706	0.7580	0.8706	0.1841	0.0339	0.1841

Table 10: Tamil: All Runs Results Ranked by Average F1 Score

Rank	Team	Run	L1 F1	L2 F1	Avg F1	L1 Acc	L1 Prec	L1 Rec	L2 Acc	L2 Prec	L2 Rec
1	Cyberpunk (Abir, 2026)	Run2	0.9638	0.6222	0.7930	0.9600	0.9698	0.9600	0.6300	0.6367	0.6300
2	CUET-2567	Single	0.9404	0.5148	0.7276	0.9600	0.9216	0.9600	0.5900	0.6124	0.5900
3	Cyberpunk (Abir, 2026)	Run1	0.9248	0.5126	0.7187	0.9200	0.9297	0.9200	0.5300	0.5286	0.5300
4	SEMANTICA	Run2	0.8861	0.5494	0.7177	0.8400	0.9680	0.8400	0.5300	0.6253	0.5300
5	INDILANGTECH (Kumar et al., 2026)	Run2	0.9248	0.4896	0.7072	0.9200	0.9297	0.9200	0.5600	0.4788	0.5600
6	SEMANTICA	Run1	0.9084	0.4978	0.7031	0.8800	0.9387	0.8800	0.4900	0.5127	0.4900
7	INDILANGTECH (Kumar et al., 2026)	Run3	0.9303	0.4302	0.6803	0.9400	0.9208	0.9400	0.4900	0.3933	0.4900
8	INDILANGTECH (Kumar et al., 2026)	Run1	0.9303	0.4243	0.6773	0.9400	0.9208	0.9400	0.5000	0.4003	0.5000
9	SYNAPSE (K P et al., 2026)	Single	0.9200	0.4256	0.6728	0.9200	0.9200	0.9200	0.5000	0.4442	0.5000
10	ALCHEMIST	Single	0.9404	0.4048	0.6726	0.9600	0.9216	0.9600	0.5000	0.4675	0.5000
11	CUETEUROPA_JOY	Run1	0.9196	0.4242	0.6719	0.9100	0.9294	0.9100	0.4200	0.4316	0.4200
11	CUETEUROPA_RATHIJIT	Run1	0.9196	0.4242	0.6719	0.9100	0.9294	0.9100	0.4200	0.4316	0.4200
13	MINDBRIDGE	Single	0.9404	0.4006	0.6705	0.9600	0.9216	0.9600	0.4700	0.3592	0.4700
14	DLRG	Single	0.9200	0.4095	0.6648	0.9200	0.9200	0.9200	0.4500	0.3921	0.4500
15	CUETEUROPA_RATHIJIT	Run3	0.9404	0.3819	0.6612	0.9600	0.9216	0.9600	0.4900	0.3740	0.4900
15	CUETEUROPA_JOY	Run3	0.9404	0.3819	0.6612	0.9600	0.9216	0.9600	0.4900	0.3740	0.4900
17	SJK	Single	0.9404	0.3742	0.6573	0.9600	0.9216	0.9600	0.4600	0.3439	0.4600
18	MEMOCRACY	Single	0.9535	0.3545	0.6540	0.9600	0.9506	0.9600	0.3800	0.4470	0.3800
19	CODECRAFTERS	Single	0.9404	0.3115	0.6260	0.9600	0.9216	0.9600	0.3500	0.3055	0.3500
20	CUETEUROPA_RATHIJIT	Run2	0.9303	0.2035	0.5669	0.9400	0.9208	0.9400	0.3300	0.4239	0.3300
20	CUETEUROPA_JOY	Run2	0.9303	0.2035	0.5669	0.9400	0.9208	0.9400	0.3300	0.4239	0.3300
21	Team_One	Single	0.8979	-	0.4490	0.8700	0.9313	0.8700	-	-	-

Table 11: Malayalam: All Runs Results Ranked by Average F1 Score

Shared Task on Depression Detection from Malayalam and Tamil Speech Data

Jyothish Lal G[1], Premjith B[1], Bharathi Raja Chakravarthi[2], Saranya Rajiakodi,[3] Durairaj Thenmozhi[4], Prasanna Kumar Kumaresan[2]

[1]Amrita School of Artificial Intelligence, Coimbatore,
Amrita Vishwa Vidyapeetham, India,
[2]Unit for Inclusive AI, Data Science Institute, University of Galway, Ireland
[3]Central University of Tamil Nadu, Thiruvarur, Tamil Nadu, India,
[4]Department of Computer Science and Engineering,
Sri Sivasubramaniya Nadar College of Engineering,
Kalavakkam, Tamil Nadu, 603110, India,

Correspondence: g_jyothishlal@cb.amrita.edu

Abstract

Depression is one of the most common mental health problems in the world. It affects a person's emotions, thinking, energy levels, and daily life. Early detection of depression is very important to provide timely support and treatment. While many studies focus on identifying depression from text, speech also carries important emotional and psychological signals that are often not fully explored. This paper presents an overview of the shared task on Depression Detection in Dravidian Languages (DD- DL). The task focuses on identifying signs of depression from speech data in two low-resource Dravidian languages: Tamil and Malayalam. Participants were provided with curated training datasets and were asked to build systems to classify speech samples as Depressed or Non-Depressed. The shared task includes two subtasks: (1) Depression detection in Tamil and (2) Depression detection in Malayalam. Participants applied various machine learning and deep learning approaches to model the acoustic and linguistic characteristics of speech. All submissions were evaluated using the macro-F1 score, which ensures fair performance measurement across classes.

1 Introduction

Depressive disorder (often referred as Depression) is a common mental disorder (Goldwaser and Aaronson, 2023). It affects the mood and thinking capability of a person. Depression is basically different from the normal mood changes or feelings that is observable in daily life, which last for few hours only (Almaghrabi et al., 2023). Basic symptoms of depression include loss of interest in activities, poor concentration, sleep interruptions, suicidal tendency, and so on, for long periods of time. As per the findings by World Health Organization (WHO), around 332 million people in the world have depression (WHO, 2025). Hence, early diagnosis and assessment of depression are improving a person's quality of life (He and Cao, 2018).

From a clinical perspective, the diagnosis of depression mainly depends on clinician judgment (Kroenke et al., 2001). But it shows limited reliability due to variability in symptoms and subjective interpretation (Jiang et al., 2017; Hong et al., 2021). Also, limited resources and lack of trained professionals affect accurate diagnosis and monitoring for depression (Almaghrabi et al., 2023). This highlights the need for objective and reliable biomarkers for depression (Low et al., 2020). In the last decade, audio and text modality have been increasingly used as reliable biomarkers in depression detection, to complement the clinical assessments (Li et al., 2025). It is observed that these modalities provide promising cues for assessing the mental conditions and emotional states of a patient in connection with depression.

Studies have shown that depressive person exhibit distinct linguistic patterns and can therefore be extracted using suitable Natural language Processing (NLP) techniques for better assessment (Rathner et al., 2018; Trifu et al., 2017; Leis et al., 2019). Similarly, the speech data acquired from a speaker or patient carries rich information about the emotive/psychological and physiological state (Cummins et al., 2015). Since speech is the natural way of communication and can be collected non-invasively, automatic analysis of voice also offers a promising route for depression detection.

The acoustic analysis of depressed speech has

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 114–123
July 3, 2026 ©2026 Association for Computational Linguistics

focused on several categories of bio-acoustic features. This include voice source measures, spectral features, prosodic parameters such as pitch, intonation, speaking rate, and articulatory formants (France et al., 2000). Voice source features such as jitter and shimmer have received considerable attention. Jitter measures cycle-to-cycle variability in fundamental frequency, while shimmer measures variability in amplitude. These parameters reflect the stability of vocal fold vibration. Several studies have reported increased jitter and shimmer values in depressed individuals, particularly in those with more severe symptoms (Low et al., 2020; Quatieri and Malyska, 2012; Silva et al., 2024). However, results are not fully consistent across studies. Some have reported negative correlations or task-dependent effects. Jitter and shimmer are more reliably measured in sustained vowels or steady phonation segments, and their reliability may decrease in spontaneous or highly variable speech (Franca, 2012). Spectral features have also been widely examined. Depression has been associated with shifts in spectral energy distribution. Some studies report increased energy in higher frequency bands, while others observe reduced energy in high-frequency regions and relatively stronger low-frequency components (Kiss and Vicsi, 2017). Mel-Frequency Cepstral Coefficients (MFCCs), which are widely used in speech processing, have also been explored in depression detection. Several studies have reported that specific MFCC coefficients differ between depressed and non-depressed speakers (Wang et al., 2019; Taguchi et al., 2018; Rejaibi et al., 2022). However, the particular coefficients identified as significant vary across studies. Differences in language, speech task type, recording conditions, and participant demographics may explain these inconsistencies. Prosodic features, including pitch, intensity, speaking rate, and pause structure, are among the most intuitively relevant markers. Many studies report reduced mean pitch and reduced pitch variability in depressed individuals (Almaghrabi et al., 2023).

Building on these acoustic observations, numerous studies have applied machine learning techniques to classify depressed versus non-depressed speech and to estimate depression severity. Early approaches used classical classifiers such as logistic regression, Gaussian Mixture Models (GMM), Support Vector Machines (SVM), k-Nearest Neighbors (KNN), and Random Forests (Long et al., 2017; Espinola et al., 2021; Jiang et al., 2017). Feature extraction is commonly performed using open-source tools such as openSMILE, Praat, COVAREP, and Voicebox. Reported classification accuracies generally range from 70 % to 85 %, depending on dataset size, feature selection, and evaluation methodology. More recent research has explored deep learning architectures, including Convolutional Neural Networks (CNNs) applied to spectrograms, CNN-LSTM models, and attention-based architectures (He and Cao, 2018; Kritika A, 2025; Abhinav et al., 2026). Despite encouraging results, several challenges remain. Demographic factors such as age and gender, as well as language differences and medication status, can significantly influence acoustic measures. Most research has focused on English datasets, with limited exploration of low-resource languages, including Dravidian languages. The scarcity of publicly available, well-annotated depression speech corpora in low-resource Dravidian languages remains as a challenge. Hence, the shared task on depression detection from Malayalam and Tamil speech data, held at DravidianLangTech@ACL 2026, is a step towards advancing the depression detection research in these low-resource languages.

2 Task Description

This shared task on depression detection using speech focuses on two major Dravidian languages: Malayalam and Tamil. The task aims to encourage research on identifying depressive indicators from speech signals in low-resource linguistic settings. Accordingly, the shared task is divided into two sub-tasks as follows:

Task 1: Depression detection in Malayalam

Task 2: Depression detection in Tamil

Participants are provided with curated speech datasets for both languages. Each audio sample is labeled as either Depressed (D) or Non-Depressed (ND). The objective is to develop automatic systems that can analyze the acoustic and prosodic characteristics of the speech recordings and accurately classify them into the appropriate category.

The dataset includes controlled recordings with varying articulation patterns, enabling participants to explore handcrafted acoustic features, deep learning-based speech representations, or hybrid approaches.

Model performance will be evaluated using the macro-F1 score, a standard metric for classification tasks that ensures balanced evaluation across

Parameter	Malayalam	Tamil
Total Samples	1,888	1,534
Depressed	888	534
Non-Depressed	1,000	1,000
Train Samples	1,688	1,374
Test Samples	200	160
Unique Speakers	8	9
Depressed	3	4
Non-Depressed	5	5

Table 1: Key corpus statistics by language and class.

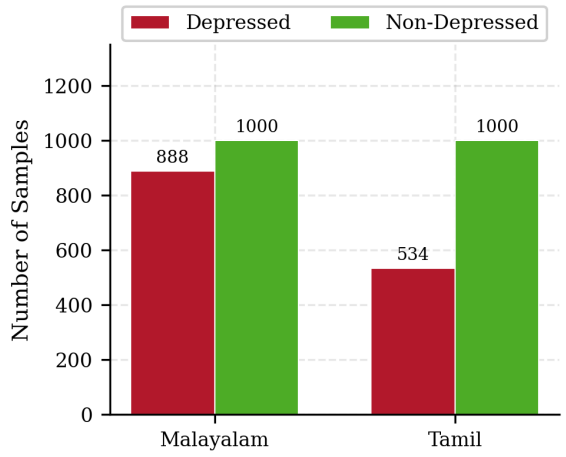


Figure 1: Sample counts by language and class (Depressed vs. Non-Depressed).

classes, particularly in scenarios with class distribution differences.

This shared task provides a benchmark platform to advance research in speech-based mental health assessment for Dravidian languages.

3 Dataset Description

Table 1 summarises the key statistics of the dataset broken down by language and class. This dataset was developed for research on automatic depression detection using speech recordings in two South Indian languages: Malayalam and Tamil. All depressed speech samples were collected in a controlled room environment with consistent acoustic settings to maintain low background noise and uniform recording quality. For the depression class, each sentence was recorded up to a maximum of four times by the same speaker with slight variations in articulation. These utterances have an average duration of 2–5 seconds and were recorded at a sampling rate of 16 kHz. In contrast, the non-depressed class consists of neutral utterances recorded at 48 kHz. In the file naming convention used, 'D' denotes a depressed sample, while any other alphabetic character (A, F, S, etc., with or without numbers) represents a unique speaker identity. Repeated versions of the same sentence are indicated at the end of the filename using numerical (1, 2, 3...) or alphabetical (a, b, c...) suffixes. Non-depressed speakers are labeled ND1, ND2, etc. The corpus comprises 3,422 total utterances across 17 unique speakers in a binary (Depressed / Non-Depressed) classification framework.

Figures 1–3 visualise the sample distribution across languages, classes, and data splits.

4 System Description

This section discusses the descriptions of the systems submitted by the participating teams. There are 62 registrations for the shared task. However, 6 teams submitted their predictions in the presceibed format through the provided Google form.

4.1 SERENE

In this work, the team tackle depression detection as a supervised binary classification problem using both text and audio approaches across Tamil and Malayalam. They employ a fine-tuned XLM-RoBERTa-base model for text analysis, trained on transcribed speech to identify depressive language patterns. For audio analysis, they utilize two methods: extracting 33 low-level acoustic features with Librosa for traditional classifiers and converting raw audio into Mel-spectrograms to train a CNN for learning time–frequency patterns linked to depression. The models are trained on labeled audio samples, predicting binary labels for unseen cases. This combination of textual and acoustic modeling effectively captures linguistic and paralinguistic indicators of depression.

4.2 TriVector

They explore speech-based depression detection through a binary classification approach that integrates handcrafted acoustic features with self-supervised speech representations. Audio recordings are standardized to 16 kHz prior to training two models in parallel: one utilizing Mel-Frequency Cepstral Coefficients (MFCCs) with a lightweight neural classifier, and the other using a fine-tuned Wav2Vec 2.0 architecture for capturing contextual speech representations. To ensure unbiased results, they implement a speaker-independent data splitting strategy and address class imbalance

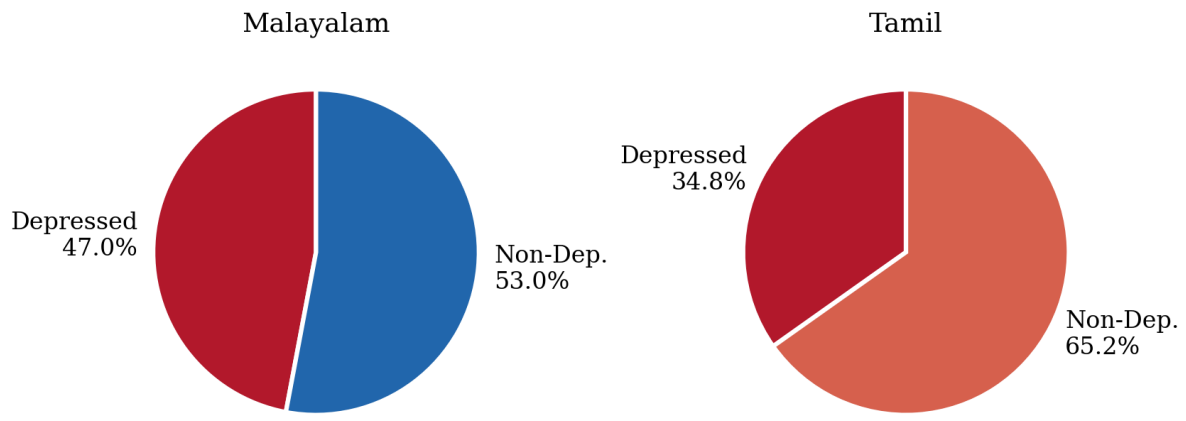


Figure 2: Class balance (Depressed / Non-Depressed) per language.

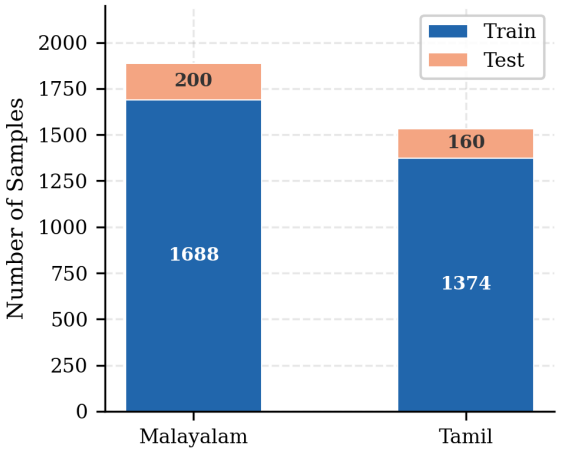


Figure 3: Train / test split per language.

via weighted loss functions. Overfitting is mitigated through techniques such as encoder freezing and early stopping. Their inference combines the outputs of both models via a weighted ensemble determined by validation F1 scores, with final predictions assessed using macro-F1 as the key metric.

4.3 Cuet_Neural_Navigators

The submission utilizes a structured machine learning pipeline for detecting depression from speech using transformer-based self-supervised audio representations. Speech recordings in Tamil and Malayalam were processed into compatible acoustic and prosodic features. Pretrained multilingual models such as Wav2Vec2-XLSR, HuBERT, and Whisper were fine-tuned for binary depression classification with speaker-aware data splits to minimize leakage. To counter overfitting, only upper encoder layers and classification heads were fine-tuned. Model performance was assessed using the macro F1-score. During inference, predictions were generated from unseen samples, and outputs from multiple models were combined through an ensemble strategy for enhanced robustness. The resulting system provides reliable depression predictions for each test utterance.

4.4 CREST

They present a language-agnostic speech-based framework for depression detection utilizing the pretrained Wav2Vec2 XLS-R (300M) model, processing raw audio waveforms to facilitate high-level speech representations without handcrafted features. Audio samples were resampled to 16 kHz and normalized, with a strict speaker-independent train-validation split to prevent overlap. The XLS-R encoder was frozen while a lightweight classification head was trained for binary classification, employing weighted cross-entropy loss to address class imbalance. The classifier was optimized with the Adam optimizer for enhanced convergence and to mitigate overfitting. Model evaluation focused on accuracy and F1-score, prioritizing balanced classification. The trained model was directly applied to unseen audio for predictions, outputting standardized CSV files. This methodology was effectively implemented across Tamil and Malayalam datasets, underscoring its scalability and cross-lingual applicability in speech-based mental health assessments.

4.5 KEC Launchpad

This system uses MFCC feature extraction combined with CNN-LSTM hybrid architecture. They

extract 40 MFCC coefficients plus delta features to capture vocal characteristics, then process them through convolutional layers for pattern detection and LSTM layers for temporal analysis. The model demonstrates strong cross-language depression detection capabilities, performing effectively on both Tamil and Malayalam speech datasets without language-specific modifications.

4.6 CODEX

In this work, the team implemented a speech-based classification approach. Audio recordings were preprocessed into mono at 16 kHz for uniformity, and a pretrained wav2vec 2.0 model extracted feature representations from the audio. These features were aggregated to create fixed-length vectors, which were used to train a Logistic Regression classifier on labeled depressed and non-depressed speech samples. The models, specific to each language, were then applied to unlabeled test audio for predictions, which were formatted into CSV files as per submission guidelines. System performance was evaluated using macro-averaged precision, recall, and F1-score.

5 Submission Analysis

Table 2 summarizes the approached used by the teams participated in this task.

Tables 3 and 4 present the official results of the Malayalam and Tamil tracks. Team SERENE achieves perfect scores across both languages, which shows the flawless classification and complete precision–recall balance. In contrast, CODEX shows substantial precision–recall imbalance, especially in Malayalam, where high precision but very low recall indicates a conservative classifier that misses many depressed cases.

SERENE achieved perfect F1 on both tracks, which demonstrates complete language-agnosticism in its system design. This likely reflects the use of multilingual pre-trained speech representations (e.g., MMS, XLS-R, or multilingual HuBERT) combined with minimal language-specific fine-tuning.

Three systems showed meaningfully higher performance on Tamil than on Malayalam. CREST exhibited the largest Tamil advantage: +0.1258 Macro F1 points (Tamil F1 = 0.9875 vs. Malayalam F1 = 0.8617). CODEX showed a +0.2324 Tamil advantage (0.5116 vs. 0.2792). KEC Launchpad achieved a marginal Tamil advantage.

TriVector and Cuet_Neural_Navigators both demonstrated higher performance on Malayalam. TriVector's Malayalam F1 (0.9950, Run 1) exceeded its Tamil F1 (0.8861, Run 2) by 0.1089 points - the largest Malayalam advantage among all teams. Cuet_Neural_Navigators showed a smaller but consistent Malayalam advantage of 0.0803 points (0.9345 vs. 0.8542).

Table 5 summarizes the descriptive statistics of Macro F1 scores for both language tracks. The mean Macro F1 for Malayalam is 0.8442 versus 0.8732 for Tamil, which indicates that the Tamil track, on average, was more tractable for participating systems by approximately 0.029 F1 points. The median Malayalam F1 is 0.9648 (between TriVector and Cuet_Neural_Navigators) versus 0.9368 for Tamil (between CREST and TriVector). The inversion of mean and median ordering between languages reflects different outlier structures: Malayalam has a single very low-performing outlier (CODEX at 0.2792) that pulls the mean down substantially, while Tamil has two outliers dragging it down but from a higher floor.

The standard deviation of F1 scores is significantly higher for Malayalam (0.2704) compared to Tamil (0.1900). This larger spread in Malayalam confirms greater inter-system variability and suggests that the Malayalam task created a harder differentiation challenge - some systems excelled while others failed substantially. The range (max-min) of 0.7208 in Malayalam versus 0.4884 in Tamil reinforces this finding.

Both tracks exhibit a ceiling behavior: the maximum F1 is 1.0 in both cases. However, the floor differs: 0.2792 for Malayalam versus 0.5116 for Tamil. The raised Tamil floor implies that even the weakest valid system managed above-chance classification on Tamil, whereas the weakest Malaysian system performed near or below chance-level for the positive class.

In addition to the metrics, discussed above, we computed the absolute F1–Accuracy divergence (|F1-Acc|) and Precision–Recall gap. The |F1-Acc| captures the extent to which a system's accuracy is being inflated by predicting the majority class more frequently. A large gap indicates systematic class bias in the prediction. The Precision–Recall gap reveals which type of error a system preferentially commits: a large positive gap means the system over-predicts the negative (non-depressed) class, achieving high precision at the cost of missing many depressed individuals (high false negatives);

Team	Ta/Ml	Fusion / Approach	XLM-R	W2V2	XLS-R	HuBERT	Whisper	MFCC	CNN-LSTM
SERENE	✓	Late fusion (text + audio)	✓	✗	✗	✗	✗	✗	✗
TriVector	✓	Parallel audio models; weighted ensemble	✗	✓	✗	✗	✗	✓	✗
Cuet_Neural_Navigators	✓	Multi-SSL fine-tuning; ensemble	✗	✗	✓	✓	✓	✗	✗
CREST	✓	Frozen XLS-R encoder + classifier head	✗	✗	✓	✗	✗	✗	✗
KEC Launchpad	✓	MFCC-based deep hybrid model	✗	✗	✗	✗	✗	✓	✓
CODEX	✓	W2V2 feature extraction + Logistic Regression	✗	✓	✗	✗	✗	✗	✗

Table 2: Comparison of submitted systems for depression detection.

Rank	Team	Run	Macro F1	Accuracy	Macro Prec.	Macro Recall
1	SERENE	Run 1	1.0000	1.0000	1.0000	1.0000
2	KEC Launchpad	Run 1	0.9950	0.9950	0.9951	0.9949
3	TriVector	Run 1	0.9950	0.9950	0.9949	0.9951
4	Cuet_Neural_Navigators	Run 2	0.9345	0.9350	0.9435	0.9337
5	CREST	Run 1	0.8617	0.8650	0.8953	0.8622
6	CODEX	Run 1	0.2792	0.2900	0.5621	0.1926

Table 3: Official results of Malayalam track

Rank	Team	Run	Macro F1	Accuracy	Macro Prec.	Macro Recall
1	SERENE	Run 1	1.0000	1.0000	1.0000	1.0000
1	KEC Launchpad	Run 1	1.0000	1.0000	1.0000	1.0000
2	CREST	Run 1	0.9875	0.9875	0.9878	0.9875
3	TriVector	Run 2	0.8861	0.8875	0.9082	0.8875
4	Cuet_Neural_Navigators	Run 2	0.8542	0.8562	0.8775	0.8562
5	CODEX	Run 1	0.5116	0.6312	0.6667	0.4208

Table 4: Official results of Tamil track

a negative gap means the system is more aggressive in flagging depression, achieving higher recall at the cost of some false alarms. CODEX team's submission in Tamil track is the most extreme case. Its accuracy of 0.6312 is 11.96 percentage points higher than its F1 of 0.5116. This divergence of 0.1196 represents an accuracy inflation of 23.4% relative to F1, which means accuracy overstates the true system quality by nearly a quarter.

The Figures 4 and 5 show radar charts for Malayalam and Tamil to plot four classification metrics (Macro F1, Accuracy, Precision, and Recall) for the top five systems. The radial scale is deliberately bounded between 0.80 and 1.00 to make differences between competitive systems visible. CODEX is excluded from both plots as a statistically significant outlier (Z = -2.09 on Malayalam, Z = -1.90 on Tamil), whose low scores would collapse the scale and obscure the performance profiles of the other teams. Each team appears as a coloured polygon: a perfectly balanced system produces a symmetric diamond pressing against the outer edge, while any inward recession on a particular axis reveals an imbalance in that metric.

In Malayalam track, SERENE and KEC Launchpad form near-perfect outer-boundary diamonds, reflecting their exceptional balance across all four metrics (F1 = 1.00 and 0.9950 respectively) with essentially no precision-recall gap. TriVector overlaps heavily with KEC Launchpad and is a statistical tie. Cuet_Neural_Navigators shows the first visible asymmetry - a slight pull toward the recall axis from mild precision dominance (gap = +0.0098), meaning it misses slightly more true cases than it raises false alarms. CREST has the most distorted polygon: precision (0.895) visibly exceeds recall (0.862), corresponding to a 13.78% false negative rate — nearly one in seven depressed speakers would be missed. Crucially, F1 and Accuracy are nearly perfectly aligned for all five systems, which confirms that none is gaming accuracy through class-imbalanced predictions.

In Tamil task, SERENE and KEC Launchpad again achieve perfect outer-boundary diamonds (all metrics = 1.00). The most striking change from the Malayalam plot is CREST: its asymmetric shape on Malayalam transforms into a near-perfect diamond on Tamil (F1 = 0.9875, P-R gap = 0.0003), which captures its dramatic rank jump from 5th on Malayalam to 2nd on Tamil. TriVector's polygon moves in the opposite direction - it becomes precision-leaning on Tamil (precision = 0.908, recall = 0.888, gap = +0.021), reversing its recall-dominant shape from Malayalam. This reversal is an architectural signal: the model makes qualitatively different errors depending on the language.

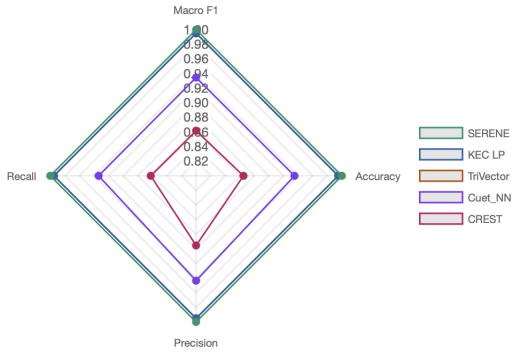


Figure 4: Radar (spider) charts showing the per-metric performance profiles of the top-five systems on the Malayalam track

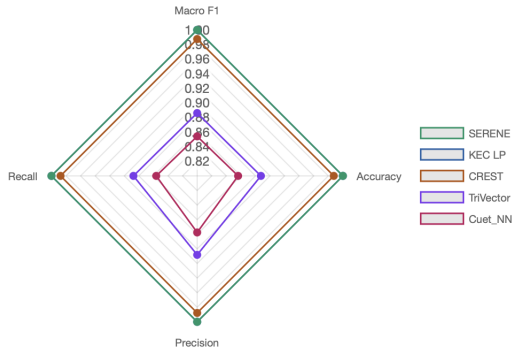


Figure 5: Radar (spider) charts showing the per-metric performance profiles of the top-five systems on the Tamil track

Table 5 shows the descriptive statistics over Macro F1-scores for all valid systems in each track reveal the overall difficulty level, performance spread, and central tendency of the competition. The mean Macro F1 for Malayalam is 0.8442 versus 0.8732 for Tamil, which indicates that the Tamil track, on average, was more tractable for participating systems by approximately 0.029 F1 points. The median Malayalam F1 is 0.9648 (between TriVector and Cuet_Neural_Navigators) versus 0.9368 for Tamil (between CREST and TriVector). The inversion of mean and median ordering between languages reflects different outlier structures: Malayalam has a single very low-performing outlier (CODEX at 0.2792) that pulls the mean down substantially, while Tamil has two outliers dragging it down but from a higher floor. The standard devia-

Statistic	Malayalam (F1)	Tamil (F1)
Mean	0.8442	0.8732
Median	0.9647	0.9368
Std Dev	0.2574	0.1715
Min	0.2792	0.5116
Max	1.0000	1.0000
Range	0.7208	0.4884

Table 5: Descriptive Statistics of Macro F1-scores by Language Track

tion of F1 scores is notably higher for Malayalam (0.2704) compared to Tamil (0.1900). This larger spread in Malayalam confirms greater inter-system variability and suggests that the Malayalam task created a harder differentiation challenge - some systems excelled while others failed substantially. The range (max-min) of 0.7208 on Malayalam versus 0.4884 on Tamil reinforces this finding. Both tracks exhibit ceiling behavior: the maximum F1 is 1.0 in both cases. However, the floor differs: 0.2792 for Malayalam versus 0.5116 for Tamil. The raised Tamil floor implies that even the weakest valid system managed above-chance classification on Tamil, whereas the weakest Malaysian system performed near or below chance-level for the positive class.

6 Conclusion

This paper presented an overview of the Shared Task on Depression Detection in Dravidian Languages at DravidianLangTech@ACL 2026. The task focused on binary classification of depression from speech data in two low-resource languages: Malayalam and Tamil. A total of 3,422 utterances from 17 speakers were released across both tracks, and system performance was evaluated using macro-F1 to ensure balanced assessment across classes. The shared task attracted 62 registrations, with six teams submitting valid runs. The participating systems explored a diverse range of approaches, including handcrafted acoustic features (MFCCs), CNN-LSTM hybrid models, pretrained self-supervised speech representations (Wav2Vec2, XLS-R, HuBERT, Whisper), and multimodal fusion combining text and audio. The results show that self-supervised and multilingual pretrained models are highly effective for this task. Both tracks exhibited strong top-end performance, with multiple systems achieving near-perfect or perfect macro-F1 scores. However, descriptive statistics reveal meaningful differences between languages. The Tamil track showed slightly higher mean performance and a higher performance floor, while the Malayalam track displayed greater inter-system variability and a wider performance range. These findings suggest that Malayalam posed a more sensitive differentiation challenge, where system design choices had a stronger impact on outcomes.

7 Limitations

This section discusses the limitations of this shared task. Here, the dataset size is relatively small, consisting of 3,422 utterances collected from only 17 speakers, which limits speaker diversity and may affect the generalizability of the developed models. Models trained on such a limited set of speakers may capture speaker-specific characteristics rather than robust depression-related speech patterns. Additionally, the task formulation is limited to binary classification (depressed vs. non-depressed) and does not account for different levels of depression severity that are relevant in clinical assessment.

8 Ethical Considerations

The dataset used in this shared task consists of elicited speech recordings rather than speech collected from clinically diagnosed depression patients. As a result, the data collection process does not involve sensitive clinical information, personal health records, or vulnerable patient groups. The recordings were obtained from speakers who produced predefined utterances in controlled settings, which minimizes potential privacy risks and ethical concerns typically associated with mental health datasets. Since no real patient data or diagnostic information was collected, the dataset avoids many ethical challenges related to informed consent, confidentiality, and the handling of sensitive medical data. However, it is important to acknowledge that systems developed using such elicited data should be interpreted carefully and should not be directly used for clinical diagnosis without proper validation using ethically collected clinical datasets.

Acknowledgments

This work was conducted with the financial support from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289_P2(Insight_2), supported in part of Science Foundation Ireland Centre for Research Training in Artificial Intelligence under Grant No. 18/CRT/6223.

References

Vura Abhinav, Bhaswanth Reddy Indukuri, MS Karthik, Sai Praneeth Reddy Alavalapati, Ramisetty Lakshmi Venkat, and G Jyothish Lal. 2026. Vision transformer-based audio analysis for depression detection: A human factor in reliable cps. In *Reliability in Cyber-Physical Systems: The Human Factor Perspective*, pages 65–81. Springer.

Shaykhah A. Almaghrabi, Scott R. Clark, and Mathias Baumert. 2023. Bio-acoustic features of depression: A review. *Biomedical Signal Processing and Control*, 85:105020.

Nicholas Cummins, Stefan Scherer, Jarek Krajewski, Sebastian Schnieder, Julien Epps, and Thomas F Quatieri. 2015. A review of depression and suicide risk assessment using speech analysis. *Speech communication*, 71:10–49.

Caroline Wanderley Espinola, Juliana Carneiro Gomes, Jessiane Mônica Silva Pereira, and Wellington Pinheiro Dos Santos. 2021. Detection of major depressive disorder using vocal acoustic analysis and machine learning—an exploratory study. *Research on Biomedical Engineering*, 37(1):53–64.

Maria Claudia Franca. 2012. Acoustic comparison of vowel sounds among adult females. *Journal of Voice*, 26(5):671–e9.

Daniel Joseph France, Richard G Shiavi, Stephen Silverman, Marilyn Silverman, and MJItoBE Wilkes. 2000. Acoustical properties of speech as indicators of depression and suicidal risk. *IEEE transactions on Biomedical Engineering*, 47(7):829–837.

Eric L. Goldwaser and Scott T. Aaronson. 2023. *Depressive Disorders*, pages 531–567. Springer International Publishing, Cham.

Lang He and Cui Cao. 2018. Automated depression analysis using convolutional neural networks from speech. *Journal of biomedical informatics*, 83:103–111.

Ran Ha Hong, Jill K Murphy, Erin E Michalak, Trisha Chakrabarty, Zuowei Wang, Sagar V Parikh, Larry Culpepper, Lakshmi N Yatham, Raymond W Lam, and Jun Chen. 2021. Implementing measurement-based care for depression: practical solutions for psychiatrists and primary care physicians. *Neuropsychiatric Disease and Treatment*, pages 79–90.

Haihua Jiang, Bin Hu, Zhenyu Liu, Lihua Yan, Tianyang Wang, Fei Liu, Huanyu Kang, and Xiaoyu Li. 2017. Investigation of different speech types and emotions for detecting depression using different classifiers. *Speech Communication*, 90:39–46.

Gábor Kiss and Klára Vicsi. 2017. Mono-and multilingual depression prediction based on speech processing. *International Journal of Speech Technology*, 20(4):919–935.

Arya Palackal Shijish Riya Rajeev Jyothish Lal G Kritika A, Meenakshy S. 2025. Dravimood: Speech-based depression classification in dravidian languages using feature fusion and deep learning. In *Proceedings of the Fourth International Conference on Speech and Language Technologies for Low-Resource Languages (SPELLL 2025)*.

Kurt Kroenke, Robert L Spitzer, and Janet BW Williams. 2001. The phq-9: validity of a brief depression severity measure. *Journal of general internal medicine*, 16(9):606–613.

Angela Leis, Francesco Ronzano, Miguel A Mayer, Laura I Furlong, and Ferran Sanz. 2019. Detecting signs of depression in tweets in spanish: behavioral and linguistic analysis. *Journal of medical Internet research*, 21(6):e14199.

Yuxin Li, Sinchana Kumbale, Yanru Chen, Tanmay Surana, Eng Siong Chng, and Cuntai Guan. 2025. Automated depression detection from text and audio: A systematic review. *IEEE Journal of Biomedical and Health Informatics*.

Hailiang Long, Zhenghao Guo, Xia Wu, Bin Hu, Zhenyu Liu, and Hanshu Cai. 2017. Detecting depression in speech: Comparison and combination between different speech types. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1052–1058. IEEE.

Daniel M Low, Kate H Bentley, and Satrajit S Ghosh. 2020. Automated assessment of psychiatric disorders using speech: A systematic review. *Laryngoscope investigative otolaryngology*, 5(1):96–116.

Thomas F Quatieri and Nicolas Malyska. 2012. Vocal-source biomarkers for depression: A link to psychomotor activity. In *Interspeech*, volume 2, pages 1059–1062.

Eva-Maria Rathner, Julia Djamali, Yannik Terhorst, Björn Schuller, Nicholas Cummins, Gudrun Salamon, Christina Hunger-Schoppe, and Harald Baumeister. 2018. How did you like 2017? detection of language markers of depression and narcissism in personal narratives.

Emna Rejaibi, Ali Komaty, Fabrice Meriaudeau, Said Agrebi, and Alice Othmani. 2022. Mfcc-based recurrent neural network for automatic clinical depression recognition and assessment from speech. *Biomedical Signal Processing and Control*, 71:103107.

Wegina Jordana Silva, Leonardo Lopes, Melyssa Kellyane Cavalcanti Galdino, and Anna Alice Almeida. 2024. Voice acoustic parameters as predictors of depression. *Journal of Voice*, 38(1):77–85.

Takaya Taguchi, Hirokazu Tachikawa, Kiyotaka Nemoto, Masayuki Suzuki, Toru Nagano, Ryuki Tachibana, Masafumi Nishimura, and Tetsuaki Arai. 2018. Major depressive disorder discrimination using vocal acoustic features. *Journal of affective disorders*, 225:214–220.

Raluca Nicoleta Trifu, Bogdan Nemeş, Carolina Bodea-Haţegan, and Doina Cozman. 2017. Linguistic indicators of language in major depressive disorder (mdd). an evidence based research. *Journal of Evidence-Based Psychotherapies*, 17(1).

Jingying Wang, Lei Zhang, Tianli Liu, Wei Pan, Bin Hu, and Tingshao Zhu. 2019. Acoustic differences between healthy and depressed people: a cross-situation study. *BMC psychiatry*, 19(1):300.

WHO. 2025. Depressive disorder (depression). *Available in:< https://www.who.int/news-room/fact-sheets/detail/depression>(accessed 19.12.2025).*

Shared Task on Prompt Style Recovery for Large Language Models in Telugu

Premjith B[1], Jyothish Lal G[1], Bharathi Raja Chakravarthi[2], Saranya Rajiakodi[3], Durairaj Thenmozhi,[4], Ratnavel Rajalakshmi,[5], Rahul Ponnusamy[2], Chinthala Bhuvanesh[1],

[1]Amrita School of Artificial Intelligence, Coimbatore,
Amrita Vishwa Vidyapeetham, India,
[2]Unit for Inclusive AI, Data Science Institute, University of Galway, Ireland,
[3]Central University of Tamil Nadu, Thiruvarur, Tamil Nadu, India,
[4]Department of Computer Science and Engineering,
Sri Sivasubramaniya Nadar College of Engineering,
Kalavakkam, Tamil Nadu, 603110, India,
[5]School of Computer Science and Engineering,
Vellore Institute of Technology, Chennai, India

Correspondence: b_premjith@cb.amrita.edu

Abstract

This paper presents an overview of the Shared Task on Prompt Recovery for Large Language Models (LLMs) in Telugu, organized as part of DravidianLangTech @ ACL 2026. The task focuses on identifying the underlying communicative style of Telugu text excerpts, framed as a nine-class single-label classification problem covering Formal, Informal, Optimistic, Pessimistic, Humorous, Serious, Inspiring, Authoritative, and Persuasive tones. The dataset was constructed by collecting Telugu YouTube comments and generating style-modified variants using an LLM, resulting in 3,000 training instances, 300 validation samples, and 301 test samples. A total of 52 teams registered for the shared task, with 13 teams submitting valid system predictions. Systems explored diverse approaches, including transformer-based fine-tuning (IndicBERT, MuRIL, XLM-R), ensemble and stacking methods, pairwise modeling strategies, curriculum learning, and few-shot large language model prompting. Evaluation was conducted using Macro F1-score as the primary metric. The top-performing system achieved a Macro F1-score of 0.2987. Overall results indicate that Telugu prompt-style recovery remains a challenging problem, particularly due to stylistic overlap and high lexical similarity across classes.

1 Introduction

Large Language Models (LLMs) are capable of producing fluent, coherent, and contextually appropriate multilingual text across a wide range of domains. Typically, LLMs are general purpose models, which are pretrained on large-scale diverse corpora. This helps LLMs to perform tasks such as summarization, translation, question answering, dialogue generation and content creation with minimal task-specific supervision.

A key factor behind the effective use of LLMs is prompt engineering, which is an approach to design and structure the input prompts to guide the LLM to obtain desired results. unlike fine-tuning, Prompt engineering (Boonstra, 2025), (Marvin et al., 2024) doesn't update the parameters of the model, but leverages carefully crafted instructions, contextual cues, examples (few-shot prompting) and formatting strategies to control the model behavior. Therefore, it is possible to influence the style and intent of generated responses through variations in wording, constraints, or demonstrations (Liu et al., 2024a), (Zarra and Chiheb, 2025).

Prompt engineering for style transfer focuses specifically on controlling the communicative tone, register, or rhetorical framing of generated text (Luo et al., 2023), (Kong et al., 2025), (Lai et al., 2024). By explicitly specifying stylistic attributes—such as formal vs. informal tone, optimistic vs. pessimistic outlook, humorous vs. serious expression, or persuasive vs. authoritative voice—prompts can direct LLMs to rewrite or generate content in a target style while preserving the underlying semantic content (Liu et al., 2024b). Effective style-oriented prompting requires sensitivity to linguistic markers such as lexical choice, sentence structure, discourse patterns, and pragmatic cues. Consequently, understanding and recovering stylistic intent becomes essential not only for generating style-consistent outputs but also for

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 124–133
July 3, 2026 ©2026 Association for Computational Linguistics

building systems capable of recognizing and modeling nuanced communicative variation across languages.

Building upon the growing research for style-aware prompting and generation, we introduce the Shared Task on Prompt Recovery for LLM in Telugu, organized as part of DravidianLangTech @ ACL 2026. The shared task focuses on Telugu Prompt-Style Recovery, where the objective is to automatically identify the communicative style underlying a Telugu transcript excerpt. Given an input sentence or short passage, participating systems must classify the text into one of nine stylistic categories: Formal, Informal, Optimistic, Pessimistic, Humorous, Serious, Inspiring, Authoritative, or Persuasive. Each category represents a distinct tonal, pragmatic, or rhetorical intention expressed by the speaker or writer (Gao et al., 2024), (Chen et al., 2024) .

The shared task attracted 52 registrations, of which 13 teams successfully submitted system predictions. We used macro F1 score for evaluating the predictions and the top performing team achieved an F1 score of 0.2987.

2 Task Description

This shared task focuses on identifying the communicative style embedded in a Telugu text excerpt. The primary objective of the task is multi-class style classification, where systems are required to predict the underlying prompt style used to generate or rewrite a given Telugu sentence or short paragraph.

Given an input text, participating systems must classify it into one of the following nine stylistic categories:

- **Formal**: polite, structured language; professional register; full sentences; minimal slang.
- **Informal**: conversational tone; colloquial expressions; slang, contractions, emojis; second-person address.
- **Optimistic**: positive outlook; encouragement; future-oriented success framing.
- **Pessimistic**: negative or doubtful tone; cautionary language; bleak outlook.
- **Humorous**: playful exaggeration; irony; jokes; light-hearted metaphors.
- **Serious**: sober and factual tone; grave subject matter; absence of humor.
- **Inspiring**: motivational language; calls to action; uplifting framing ("you can do it").
- **Authoritative**: directive or commanding voice; expert-like certainty; imperative guidance.
- **Persuasive**: convincing or benefit-oriented appeals; lobbying or sales-like language.

Dataset	No. of Instances
Train	3,000
validation	300
Test	301

Table 1: The distribution of the dataset across train, validation and test datasets

The problem is formulated as a single-label multi-class classification task, where each instance belongs to exactly one style category. Systems are evaluated using standard classification metrics to ensure fair comparison and reproducibility.

3 Dataset Description

The dataset was prepared by collecting comments in Telugu from YouTube. The comments were passed to ChatGPT[1] to generate style-modified versions. Each instance of the dataset consists of an original Telugu text paired with a style-modified version.

The dataset was split into three - train, test and validation. The split is given in Table 1. Each dataset contains four columns - ID (A unique identifier for each instance), original transcripts (Contains the original, unmodified Telugu text), Change style (Contains the modified version of the original transcript), and STYLE (Specifies the target stylistic category).

The *ORIGINAL TRANSCRIPTS* column contains the base Telugu sentences, while the *CHANGE STYLE* column provides modified versions of those sentences in a different stylistic tone. The *STYLE* column represents the target label, consisting of 9 distinct style categories. All nine styles are present in the training, validation, and test splits. In the training set, the most frequent style is Pessimistic with 347 samples; in the validation set,

[1] chatgpt.com

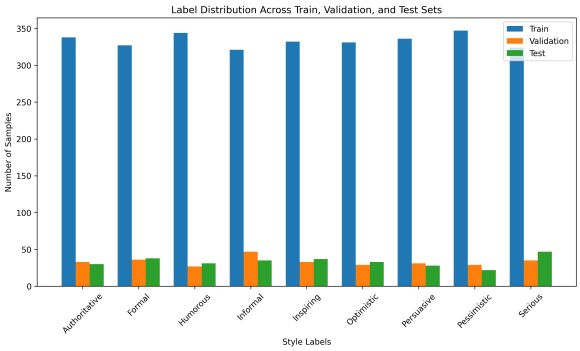


Figure 1: The distribution of data across nine classes in train, validation and test datasets

Informal appears most frequently with 47 samples; and in the test set, Serious is the most frequent with 47 samples. This suggests a reasonably distributed multi-class setting without severe imbalance.

The training set contains 395 unique original transcripts and 2545 unique style-modified sentences, indicating substantial paraphrasing variation in the CHANGE STYLE column. The validation and test sets also show strong diversity, with over 200 unique original transcripts in each split and nearly all style-modified sentences being unique. This demonstrates that the dataset supports robust modeling for paraphrasing, stylistic transformation, and embedding-based learning. The data are evenly distributed across nine classes in train, validation and test datasets. The label distribution is illustrated in Figure 1. Among all the classes, Pessimistic (347 samples), Informal (47 samples) and Serious (47 samples) are the most frequent classes in training, validation and test datasets.

Below is an example instance used for this task.

Original text — వంశీ అనేది ఒక విద్యార్థి అతనికి డిగ్రీ ఉంది ఉద్యోగాలు స్కిల్ స...ాలు కావాలంటే తెలపండి లేదా సారాంశం మాత్రమే అవసరమైతే అనుమతించండి.

Changed style — ఈ రోజుల్లో విద్యార్థులు విద్యతో పాటు వివిధ నైపుణ్యాలను అభ్యసి...టికల్ అనుభవం ఇవే మీ ఉద్యోగ అవకాశాలను గణనీయంగా పెంచుతాయి.

Prompt style - Formal

4 System Description

This section discusses the descriptions of the systems submitted by the participating teams. There are 52 registrations for the shared task. However, 11 teams submitted their predictions through the provided Google form.

4.1 Still Loading

Team Stll Loading's submissions demonstrate a hybrid late-fusion ensemble that combines semantic grounding specific to Indic language with general multilingual capabilities. The system successfully captures both global semantic patterns and the unique morphosyntactic features of the Telugu script by fusing models like XLM-RoBERTa Large (Conneau et al., 2020) and mBERT (Devlin et al., 2019) with specialized frameworks like ai4bharat/IndicBERTv2-MLM-only (Doddapaneni et al., 2023) and MuRIL (Khanuja et al., 2021). For cross-segment style grounding, the method involves concatenating the styled target text with the original context. An optimization pipeline that combines label smoothing, cosine learning rate scheduling, and gradient accumulation achieves robust performance across a variety of Telugu "Rasas" while keeping an effective batch size of 32 constant.

4.2 Mano_sub

The MuRIL model was trained using a "frozen-backbone" technique, in which all transformer layers were frozen and only the classification head was updated. The AdamW optimizer (Loshchilov and Hutter, 2017) was used for 10 training epochs with a batch size of 16 and a learning rate of 1×10^{-4}. They used the Metal Performance Shaders backend for GPU acceleration on macOS. A sliding window with a 256-token sequence length was employed for inference. To ensure that the classifier produced results that were generally applicable, final predictions for the unlabeled test set were based on the epoch that obtained the highest Macro F1-score.

4.3 Medhastra

Team Medhastra uses XLM-RoBERTa-base to reformulate the Telugu prompt style classification task into a pairwise binary classification problem. By concatenating the input text with style labels, they produce nine candidate pairs for every training instance, increasing the training set from 3,000 to 27,000 instances in spite of the scarcity of labeled data. With one positive and eight negative examples per instance, this method produces a balanced binary task. The binary classifier evaluates

each style candidate during inference and chooses the one with the highest confidence score. After four epochs of fine-tuning, the model achieved a macro F1 score of 0.4915, with training and validation loss decreasing over time.

4.4 DLRG

The proposed system combines three different models - XLM-R, mT5 (Xue et al., 2021) and BiLSTM instead of relying on just one. By merging their prediction scores through a simple stacking method, the system uses each model's strengths, reduces errors, and achieves better and more stable accuracy. Stacked interaction and entropy features train a calibrated Logistic Regression meta-classifier, combining model strengths to improve robustness and macro-F1 performance.

4.5 JerinWarriors

They used a model that compares the original text and the rewritten text together. Both texts are processed at the same time so the model can learn how the style changes. The model focuses on the rewritten text and predicts the final style label based on the differences it learns. The model was trained using labeled examples of original and rewritten texts. Texts were limited to the first part to save memory and improve learning. During testing, when the original text was not available, the rewritten text was used in both inputs so the model could still make predictions.

4.6 TechNova

First, the training and validation datasets were combined, missing data was eliminated, and label formats were standardized across nine stylistic categories as part of a systematic preprocessing step for the proposed system. To guarantee proportional representation in both splits, the cleaned corpus was stratified. To find lexical patterns and stylistic cues in Telugu, feature extraction employed TF-IDF vectorizations based on word n-grams (one to four tokens). To deal with class imbalances and noisy annotations, an ensemble classifier that combined a Linear Support Vector machine, a Logistic regression model, and a Multinomial Naive Bayes model worked within a hard voting framework.

4.7 DeepScope

Using pretrained Indic language models, they tackled the Telugu Prompt-Style Recovery task by presenting it as a nine-class style classification problem. Through a custom PyTorch training loop, IndicBERTv2 was fine-tuned on a curated dataset. The model input consisted of task instructions and rewritten Telugu text that did not change or reveal target labels. Validation, the removal of samples with missing or invalid text, and intra-class consistency filtering to remove noise all improved data integrity. A curriculum-based approach was used for training, starting with high-confidence samples and progressively adding more ambiguous data.

4.8 PromptRecovery_Alchemists (PR_Alchemists)

The system uses GPT-5.2 (OpenAI, 2026) as its core model and employs a large language model-based approach through the DSPy framework. It builds 27 demonstrations from the training set using original Telugu texts, their style-transformed versions, and style labels, combining few-shot in-context learning and Chain-of-Thought reasoning. Untruncated text pairs and a DSPy signature (Khattab et al., 2024), (Khattab et al., 2022) directs the comparison of tone, punctuation, sentence structure, word choice, and emotional markers are given to the model during inference. Across nine categories—Authoritative, Formal, Humorous, Informal, Inspiring, Optimistic, Persuasive, Pessimistic, and Serious—the Chain-of-Thought module enables intermediate reasoning for more understandable style analysis.

4.9 Error_500

This submission explores multiple transformer-based approaches for Telugu style classification. Specifically, it utilizes IndicBERT v2 and XLM-RoBERTa base as foundational pretrained language models. The approaches incorporate varied training strategies, preprocessing techniques, and imbalance-handling mechanisms to enhance robustness and generalization. Each model is trained and validated independently, and predictions are generated from their respective best-performing checkpoints for comparative evaluation using macro-level metrics.

4.10 codecrackers

In this submission, they utilized the mT5-small transformer model from Hugging Face to predict one of nine predefined style labels. The methodology involved framing a prompt-based classification, where each text sample was fed to the model

with a structured instruction asking it to output the corresponding numeric style label. During inference, the model generated a number per input text, which was then validated and saved as the final label.

4.11 Semantica

They define the prompt-recovery task as style identification from pairs of texts: an Original Transcript and a Changed-Style Transcript. The training set is cleaned to enhance label consistency by removing duplicates, conflicting pairs with multiple style labels, and overlaps between training and validation datasets. A sliding-window strategy is employed for fine-tuning a Transformer classifier due to the encoder's context limit, where a fixed input template concatenates both texts, maximizing length constraints while generating overlapping windows. Training utilizes windowed examples inheriting their parent label and optimizing cross-entropy loss, with a weighted sampler for class balancing.

4.12 Cuet Yet Another Baseline (CYAB)

The submission investigates Telugu stylistic tone classification using transformer-based supervised learning. It fine-tunes a pretrained language model (l3cube-pune/telugu-bert) (Joshi, 2022) on a dataset with original and style-modified transcripts to explicitly capture stylistic changes. Enhanced encoder representations combine the [CLS] token with mean pooled token embeddings, processed through a multi-layer feed-forward classification head. The training applies weighted cross-entropy loss, cosine learning rate scheduling with warmup, and early stopping based on validation macro F1 score. In another run, MuRIL-base is fine-tuned on style-transferred Telugu text, concatenating [CLS] with surface-level linguistic features and classifying through a fusion layer using class-weighted cross-entropy loss. Final predictions are derived from soft voting across three seeds.

4.13 Axiom

The system utilizes a stacking ensemble of classical machine learning models, trained on TF-IDF features from the "Change Style" column, which encodes stylistic cues. It integrates character-level n-grams (2–6) and word-level n-grams (1–3) through a TF-IDF FeatureUnion with sublinear term frequency scaling, enhancing its sensitivity to Telugu morphological patterns and phrase-level style signals. A Logistic Regression meta-learner is employed on the probability outputs of these models, trained using 5-fold cross-validation to create meta-features. Evaluation involved a stratified 80/20 training data split for validation accuracy, followed by retraining on the full dataset for predictions on the unlabeled test set.

Table 2 summarizes the fusion approaches, embedding approaches and various notable features followed by the participating teams.

5 Analysis of the Submission

In this task, systems were evaluated on a held-out test set, with submissions assessed across four metrics: Macro F1-Score (primary ranking criterion), Accuracy, Precision, and Recall. Teams were permitted to submit three runs; the best-performing run per team was used for the official ranking.

Table 3 presents the official results for all 13 participating teams, sorted by Macro F1-Score (descending).

Figure 4 illustrates the Macro F1-scores across all 13 teams. The team Error_500 leads the ranklist with an F1 score of 0.2987, but the margin over 2nd place (JerinWarriors, 0.2588) is relatively small ($\approx$0.04). From the figure it is evident that a clear performance cliff exists between Rank 5 (0.2285) and Rank 6 (0.1745). The performance of bottom 3 teams (Axiom, Mano_Sub, DeepScope) score are well below 0.10, which may be due to the weaker approaches they designed for this task. Another interesting inference is most teams score below the average. It is evident from the results that Error_500 leads with an F1 of 0.2987, nearly $10\times$ that of last-place DeepScope (0.0289). Another inference from the Table 3 is that PromptRecovery_Alchemists (Rank 4) achieves the highest precision of any team (0.3346), surpassing even the 1st-place team, but this comes at the cost of lower recall (0.2472), which results in a lower overall F1.

Figure 2 maps each team in precision-recall space, colored by F1-score. Teams near or above the diagonal (P=R line) tend to have balanced metrics; systems far from the diagonal exhibit metric imbalance which may indicate different modeling assumptions or post-processing strategies. In the figure, the X-axis and Y-axis represent recall and precision, respectively. It is clear from the figure that PromptRecovery_Alchemists (PR_Alchemists) is the biggest outlier with a high-

Team	Fusion / Approach	MuRIL	IndicBERT	XLM-R	mT5	GPT	TF-IDF	Stacking	Sliding	Pairwise	Weighted	Curriculum	Notable Feature
Still Loading	Late fusion (avg logits)	✓	✓	✓	×	×	×	×	×	×	✓	×	Label smoothing, cosine LR
Mano_sub	Frozen backbone + classifier	✓	×	×	×	×	×	×	×	×	×	×	Sliding window (256), MPS
Medhastra	Pairwise binary classification	×	×	✓	×	×	×	×	×	✓	×	×	Expands 3k→27k
DLRG	Stacking + LR meta	×	×	✓	✓	×	×	✓	×	×	×	×	Entropy features
JerinWarriors	Joint pair processing	×	×	×	×	×	×	×	×	×	×	×	Dual-input contrast
TechNova	Hard voting ensemble	×	×	×	×	×	✓	✓	×	×	×	×	Word n-grams (1–4)
DeepScope	Fine-tuned single model	×	✓	×	×	×	×	×	×	×	×	✓	Curriculum learning
PR_Alchemists	Few-shot CoT (DSPy)	×	×	×	×	✓	×	×	×	×	×	×	27-shot demos
Error_500	Multi-strategy ensemble	×	✓	✓	×	×	×	×	×	×	✓	×	Imbalance handling
codecrackers	Zero-shot prompting	×	×	×	✓	×	×	×	×	×	×	×	Numeric label gen.
Semantica	Sliding window + logit agg.	×	×	✓	×	×	×	×	✓	×	×	×	Weighted sampler
CYAB	Soft voting (3 seeds)	✓	×	×	×	×	×	×	×	✓	×	×	CLS + mean pool
Axiom	Stacking + LR meta	×	×	×	×	×	✓	✓	×	×	×	×	Char (2–6), word (1–3)

Table 2: Comparison of methodologies and architectural choices across participating teams.

Team Name	Run	F1	Accuracy	Precision	Recall	Rank
Error_500	Run 3	0.2987	0.2990	0.2979	0.3144	1
JerinWarriors	Run 1	0.2588	0.2691	0.2926	0.2812	2
DLRG	Run 1	0.2451	0.2425	0.2716	0.2496	3
PromptRecovery_Alchemists	Run 1	0.2406	0.2525	0.3346	0.2472	4
Cuet Yet Another Baseline	Run 2	0.2285	0.2193	0.2406	0.2356	5
Semantica	Run 1	0.1745	0.1960	0.1619	0.2194	6
Still Loading	Run 1	0.1703	0.1761	0.1679	0.1865	7
Codecrackers	Run 1	0.1516	0.1761	0.1716	0.1735	8
Medhastra	Run 1	0.1205	0.1196	0.1205	0.1231	9
TechNova	Run 1	0.1106	0.1462	0.0974	0.1365	10
Axiom	Run 1	0.0797	0.0797	0.0797	0.0805	11
Mano_Sub	Run 1	0.0491	0.1462	0.0326	0.1161	12
DeepScope	Run 1	0.0289	0.0498	0.0236	0.0674	13

Table 3: Official rank list for the Prompt Recovery for LLM in Telugu shared task.

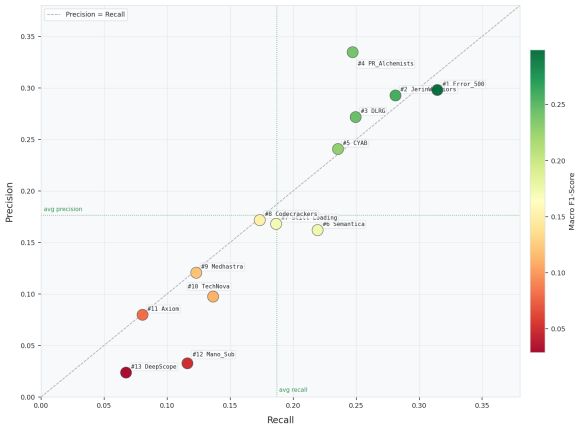


Figure 2: Precision vs Recall Scatter Plot

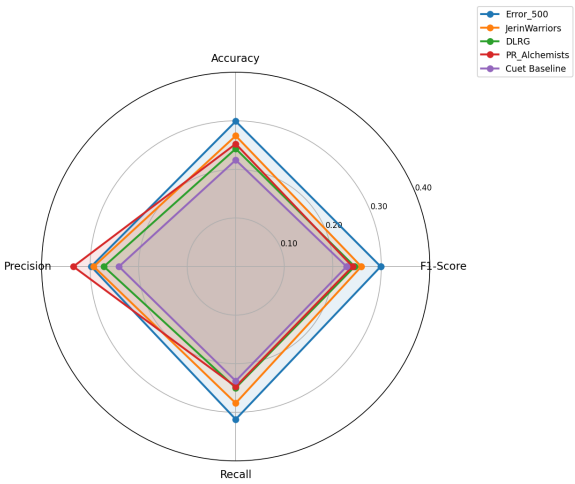


Figure 3: Radar chart for top 5 teams. A larger, more uniform polygon indicates better overall performance.

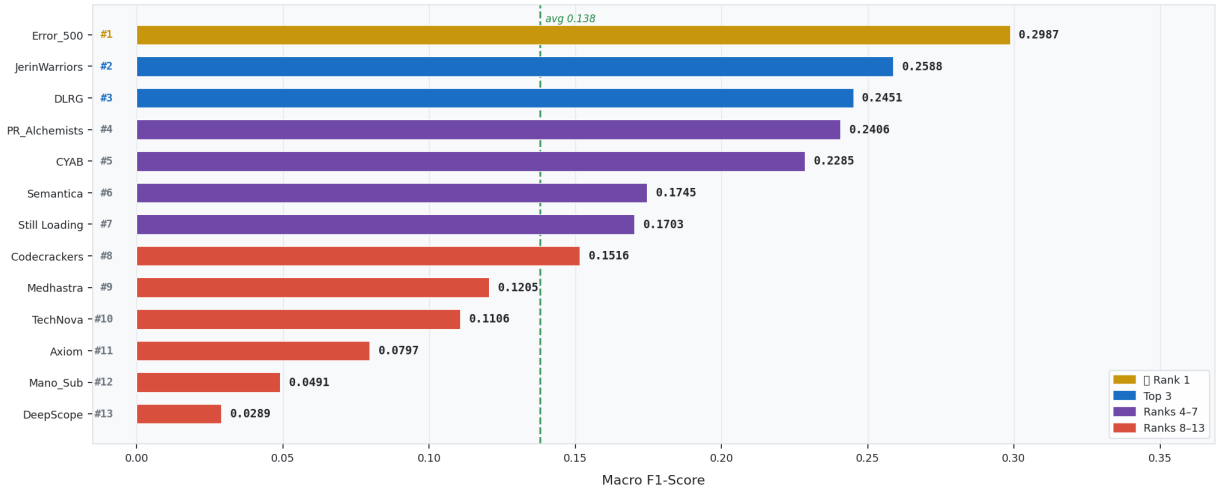


Figure 4: Rankings of the teams participated in the shared task

est precision (0.3346) but relatively low recall (0.2472). Semantica and TechNova are recall-heavy teams, who sit below the diagonal. Error_500, JerinWarriors, and DLRG cluster near the diagonal in the top-right, which indicates a balanced and well-tuned approach for the task. Mano_Sub and DeepScope appear in the bottom-left, which shows low scores on both axes, and it indicates weak overall performance from both teams.

Figure 3 presents a radar chart comparing the top 5 teams across all four metrics simultaneously, enabling a holistic comparison of their strengths and weaknesses. Error_500 has the most uniform and expansive polygon across all metrics. PromptRecovery_Alchemists has a distinctly elongated shape toward Precision, confirming its precision-biased strategy at the expense of recall.

Table 4 provides descriptive statistics for all four evaluation metrics across the 13 systems. Precision exhibits the largest range (0.3110), indicating the greatest variability in how teams trade off false positives. Recall is generally higher than precision across most teams, suggesting a mild recall bias in system design — possibly because systems over-generate candidate prompt segments to maximize recall.

All the top-performing teams scored low F1-score, which is due to the presence of same words and sentences in difference in the dataset. There are 2545 unique changed style texts in the training dataset, and 455 are repeated, which accounts to the 84.84% of the unique changed style text. The presence of the repeated contents generated by the LLMs for different style can be the reason for obtaining low scores for teams. Two examples of style modified text with similarities among multiple style tones are given in the appendix.

Statistic	F1	A	P	R
Mean	0.1379	0.1656	0.1440	0.1688
Median	0.1205	0.1761	0.1205	0.1735
Std Dev	0.0820	0.0722	0.0844	0.0726
Maximum	0.2987	0.2990	0.3346	0.3144
Minimum	0.0289	0.0498	0.0236	0.0674
Range	0.2698	0.2492	0.3110	0.2470

Table 4: Descriptive statistics of evaluation metrics. (A) Accuracy, (P) Precision, (R) Recall

Most teams show near-identical F1 and Accuracy values, suggesting a reasonably balanced label distribution in the test set. Notable exceptions (TechNova, Mano_Sub) may reflect unusual or degenerate prediction patterns.

6 Conclusion

The Prompt Style Recovery Shared Task for Telugu provides one of the first systematic evaluations of style identification in LLM-generated Telugu text. Despite diverse modeling strategies—including transformer fine-tuning, ensemble stacking, pairwise contrastive formulations, curriculum learning, and LLM-based few-shot reasoning—the overall performance across systems remained modest, with the best Macro F1-score reaching 0.2987 . The relatively low scores highlight the intrinsic difficulty of distinguishing nuanced communicative styles, especially when lexical overlap exists across categories and when style-modified texts

share substantial semantic and structural similarity. Analysis of system behaviors revealed different optimization trade-offs, with some models favoring precision while others leaned toward recall. Transformer-based and ensemble approaches generally outperformed classical baselines, suggesting that contextual semantic modeling is crucial for capturing subtle stylistic cues in Telugu. However, repeated or near-identical LLM-generated style variations within the dataset likely limited discriminative learning.

7 Limitations

This section discusses the limitations of the shared task. The first challenge is the size of the dataset, which is relatively small, consisting of only 3,000 training samples with 300 validation and 301 test instances. Such a limited dataset may restrict the ability to generalize effectively and learn robust stylistic representations. Another limitation arises from the inherent overlap between some of the stylistic categories. Classes such as optimistic, inspiring, and persuasive, or serious and authoritative, may share similar lexical and semantic characteristics. This overlap introduces ambiguity in the classification process and increases the difficulty of clearly separating stylistic boundaries. Future work could address these limitations by expanding the dataset size, incorporating more naturally occurring style variations, and exploring richer contextual modeling approaches for multilingual prompt-style recovery.

8 Ethical Considerations

The dataset used in this shared task was constructed from publicly available Telugu YouTube comments and style-modified using a large language model. Care was taken to ensure that the dataset does not contain personally identifiable information. The task focuses solely on stylistic classification and does not attempt to infer sensitive personal attributes of individuals. Nevertheless, since the dataset contains user-generated content, there may be instances of informal or subjective language. Researchers using this dataset should ensure responsible use of the data and avoid deploying models in ways that could misrepresent or misuse stylistic interpretations.

Acknowledgments

This work was conducted with the financial support from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289_P2(Insight_2), supported in part of Science Foundation Ireland Centre for Research Training in Artificial Intelligence under Grant No. 18/CRT/6223.

References

Lee Boonstra. 2025. Prompt engineering. *Google, https://www. kaggle. com/whitepaper-prompt-engineering*.

Jianlong Chen, Wei Xu, Zhicheng Ding, Jinxin Xu, Hao Yan, and Xinyu Zhang. 2024. Advancing prompt recovery in nlp: A deep dive into the integration of gemma-2b-it and phi2 models.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. 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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2023. Towards leaving no Indic language behind: Building monolingual corpora, benchmark and models for Indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12402–12426, Toronto, Canada. Association for Computational Linguistics.

Lirong Gao, Ru Peng, Yiming Zhang, and Junbo Zhao. 2024. DORY: Deliberative prompt recovery for LLM. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10614–10632, Bangkok, Thailand. Association for Computational Linguistics.

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

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.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive NLP. *arXiv preprint arXiv:2212.14024*.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. Dspy: Compiling declarative language model calls into self-improving pipelines.

Chaona Kong, Jianyi Liu, Yifan Tang, and Ru Zhang. 2025. Neuron activation modulation for text style transfer: Guiding large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7735–7747.

Wen Lai, Viktor Hangya, and Alexander Fraser. 2024. Style-specific neurons for steering LLMs in text style transfer. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13427–13443, Miami, Florida, USA. Association for Computational Linguistics.

Qingyi Liu, Jinghui Qin, Wenxuan Ye, Hao Mou, Yuxuan He, and Keze Wang. 2024a. Adaptive prompt routing for arbitrary text style transfer with pre-trained language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):18689–18697.

Shenyang Liu, Yang Gao, Shaoyan Zhai, and Liqiang Wang. 2024b. Stylerec: A benchmark dataset for prompt recovery in writing style transformation. In *2024 IEEE International Conference on Big Data (BigData)*, pages 1678–1685.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Guoqing Luo, Yu Han, Lili Mou, and Mauajama Firdaus. 2023. Prompt-based editing for text style transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5740–5750.

Ggaliwango Marvin, Nakayiza Hellen, Daudi Jjingo, and Joyce Nakatumba-Nabende. 2024. Prompt engineering in large language models. In *Data Intelligence and Cognitive Informatics*, pages 387–402, Singapore. Springer Nature Singapore.

OpenAI. 2026. Introducing gpt-5.2.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Taoufiq Zarra and Raddouane Chiheb. 2025. The influence of prompt politeness on response quality in large language models. In *2025 International Conference on Circuit, Systems and Communication (ICCSC)*, pages 1–7. IEEE.

A Examples of Style Modified Text with Similarities among Multiple Style Tones

Original text — అందరికీ నమస్కారం అండి. అందరూ క్షేమంగా ఉన్నారని తలుస్తున్నాను. మనం ఏ తరగతి నూతన పాఠ్యపుస్తకంలో భాగంగా, ఐదవ పాఠ్యాంశం జలియన్ వాలాబాగ్ గురించి రెండు భాగాల్లో గా చెప్పుకున్నాం. ఇప్పుడు మూడో, నాలుగో భాగాలను కూడా పరిగణనలోకి తీసుకుని, మరికొన్ని ప్రతిపదార్థ పద్యాలను విశ్లేషించుకుందాం. ఈ పద్యాల నుండి మీరు గొప్ప జ్ఞానాన్ని పొందవచ్చు.జలియన్ వాలాబాగ్ అమృత్సర్ పట్టణంలోని ఒక ఉద్యానవనం. ఇక్కడ ఏప్రిల్ న జరిగిన దారుణమైన సంఘటన కారణంగా బ్రిటిష్ సైనికులు అమాయకులపై కాల్పులు జరిపి అనేక మంది ప్రాణాలు కోల్పోయారు. ఈ కాల్పుల్లో మంది ఆ తీరాం మరణించారు. చాలా మంది గాయపడ్డారు. ఉమర్ ఆలీషా గారు ఈ ఘటనను తాము రాసిన ఖండకావ్యాలలో సులభమైన పద్యాలతో వివరిస్తూ, భారతీయుల సరితక ధైర్యాన్ని, నిర్భయత్వాన్ని ఆస్వాదిస్తూ, ప్రభుత్వల పీడనలపై వ్యథ వ్యక్తం చేయడమే లక్ష్యం. ఈ చరిత్రను మీరు తప్పక తెలుసుకోవాలి.పద్య భాగం: ...

Prompt styles - Serious (Train ID: PR_TE_TR_0378) and Pessimistic (Train ID: PR_TE_TR_0379)

Percentage of match - 96.42

Translation - Hello everyone. I hope everyone is well. As part of the new textbook for class 6, we have discussed the fifth chapter of Jallianwala Bagh in two parts. Now, let us consider the third and fourth parts and analyze some more anti-material poems. You can gain great knowledge from these poems. Jallianwala Bagh is a park in the city of Amritsar. Due to the brutal incident that took place here in April, British soldiers opened fire on innocent people and many people lost their lives. Many people died in this firing. Many were injured. Umar Alisha has described this incident in simple poems in his Khandakavyas, enjoying the courage and fearlessness of Indians and expressing grief over the oppression of the government. You must know this history. Verse: ...

Original text — చరిత్రకి వినీ ఎరుగని విపత్తు ఇదే. కరీబ్ గంటలే తమిళనాడు మే ఆపత్ కి బారిష్ హో రహి హై. ఐసీ బారిష్ కి సాళ్లకా రికార్డ్ టూట్ గయా హై. చెంబరంబాకం ఏరియాలో అరసం మున్నోటియే తిరకాదదే. చెన్నై మరియు పూరనగర్‌కు లేటెస్ట్ న్యూస్ కమింగ్ ఇన్ ఫ్రమ్ చెన్నై, అక్కడ హెవీ రెయిన్స్ వల్ల బ్రీఫ్ రెస్పైట్ ఉంది. బంగాళాఖాతంలో ఏర్పడిన అల్పపీడనం కారణంగా తమిళనాడు రాష్ట్రవ్యాప్తంగా విస్తారంగా వర్షాలు కురుస్తున్నాయి. తెలుగు ఫిల్మ్ ఇండస్ట్రీకి చెందిన యాక్టర్లు అందరం ఒక గ్రూప్‌గా ఫామ్ అయ్యి మద్రాస్ కోసం ఒక క్యాంపెయిన్ స్టార్ట్ చేశాం. నవంబర్‌లో చెన్నై సిటీ జనాలందరూ గుంపులు గుంపులుగా ఫామ్ అయ్యి ఇళ్లలో ఉన్న మిగతావాళ్లిని కాపాడటానికి ట్రై చేస్తున్నారు. నీళ్లు నడుము వరకు వచ్చేస్తున్నాయి. కొన్ని చోట్ల మనిషి నిలుచుంటే తల మాత్రమే కనిపిస్తుంది. కొద్దిసేపటికే వేల మంది ఇల్లు లేని వాళ్ళు అయిపోయారు. వాతావరణం కనికరిస్తే బ్రతకాల్సిన పరిస్థితికి జనం వచ్చేసారు. హారిఫిక్ సిచ్యుయేషన్ వి హవ్ బీన్ ఫాల్ ఓవర్ తమిళనాడు అండ్ పుదుచ్చేరి. తమిళనాడులో వర్షాలు, వరదల బీభత్సం కొనసాగుతోంది.....

Prompt styles - Humorous (Train ID: PR_TE_TR_0338) and Pessimistic (Train ID: PR_TE_TR_0339)

Percentage of match - 100
Translation- This is a disaster that has never been heard of in history. It has been almost an hour since the Tamil Nadu floods. The annual record of IC rainfall has been broken. In the Chembarambakkam area, there is a brief respite from heavy rains. The latest news for Chennai and Puranagar is coming in from Chennai, where there is a brief respite due to heavy rains. Due to the low pressure formed in the Bay of Bengal, heavy rains are falling across the state of Tamil Nadu. All the actors from the Telugu film industry have formed a group and started a campaign for Madras. In November, all the people of Chennai city have formed groups and are trying to save the rest of the people who are in their houses. The water is reaching the waist. In some places, only the head is visible if a person is standing. In a short time, thousands of people have become homeless. People have come to a situation where they have to survive if the weather allows. Horrific situation has fallen over Tamil Nadu and Puducherry. Rains and floods continue to wreak havoc in Tamil Nadu. Northeast cyclones are lashing Chennai.....

TamilPoliSent 2026: A Shared Task report on Multiclass Political Sentiment Analysis in Tamil

Mani Vegupatti[1], Kishore Kumar Ponnusamy[2], Bharathi Raja Chakravarthi[1], Saranya Rajiakodi[3], Durairaj Thenmozhi[4], Prasanna Kumar Kumaresan[1], Sathiyaraj Thangasamy[5]

[1]Unit for Inclusive AI, Data Science Institute, University of Galway, Ireland,
[2]Digital University Kerala, India,
[3]Central University of Tamil Nadu, India,
[4]Sri Sivasubramaniya Nadar College of Engineering, Chennai, India,
[5]Sri Krishna Adithya College of Arts and Science, India

Abstract

Political multiclass sentiment analysis involves categorizing social media comments into predefined political sentiment classes. This paper gives an overview of the TamilPoliSent 2026 shared task, which is about multiclass political sentiment analysis in Tamil. It was part of DravidianLangTech@ACL 2026. The task focused on identifying seven political sentiment categories from Tamil X (Twitter) comments in a low-resource setting. Participants were provided with annotated training and development datasets, while the test set was released without labels for final evaluation. A total of 22 teams submitted system outputs for evaluation. The submitted approaches ranged from traditional TF-IDF-based machine learning baselines to transformer fine-tuning, hybrid lexical–contextual architectures, ensemble strategies, and imbalance-aware optimization techniques. We used the macro-F1 score to check the system's performance so that all sentiment categories were fairly judged. The best-performing system achieved a macro-F1 score of 0.3935. The results highlight the challenges of modeling fine-grained political sentiment in Tamil, including class imbalance, sarcasm, informal writing styles, and semantic overlap between categories. Overall, the results show how important it is to use transformer-based models with class-balanced learning and hybrid representations for improving sentiment classification in low-resource political discourse

1 Introduction

Social media platforms such as X (formerly Twitter) have enabled the exponential rise of political comments on the World Wide Web. The ease of user participation enables real-time engagement and debates, and furthermore allows the expression of their own narratives and political sentiments. These sentiments are often multi-layered, fine-grained, and full of nuances. These characteristics necessitate sophisticated multi-class models to analyze complex political sentiments in real time and at scale. (Wankhade et al., 2022; Hermida, 2012; Velasquez, 2012; Tumasjan, 2010).

Tamil political sentiment analysis has its own challenges introduced by the nature of the language, such as rich morphology and various techniques of inflection requiring more complex processing. Moreover, the online content such as comments brings additional complexities, including code-mixing, transliteration, informal writing styles, sarcasm, implicit meanings and high noise levels making classification tasks more difficult. In addition, a low-resource setting introduces more layers of complexities, including limited annotated political data and the presence of class imbalances which impact the effectiveness of machine learning tasks. (Manoj, 2024; Shanmugavadivel, 2022; Mahata et al., 2020; Anbukkarasi and Varadhaganapathy, 2020; Thavareesan and Mahesan, 2019).

To facilitate the progress of machine learning models development in this area, the 'Political Multiclass Sentiment Analysis of Tamil X comments' shared task was introduced in 2025 (Chakravarthi et al., 2025). This task provided the participants with a 'Multi-class Tamil political sentiment analysis dataset' with seven classes - Substantiated, Sarcastic, Opinionated, Positive, Negative, Neutral, and None of the above for experimenting various techniques, building the models and benchmarking their results.

Building on the success of the 2025 shared task (Chakravarthi et al., 2025), we introduced the 2026 (Vegupatti et al., 2026) shared task to tap recent

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 134–143
July 3, 2026 ©2026 Association for Computational Linguistics

advancements in NLP over the last year and further improve the performance in the Tamil political multiclass sentiment analysis task and report the new benchmarks. The 2026 task attracted 22 participating teams, who explored a wide range of approaches, and their methodologies and details are presented in the subsequent sections.

2 Related Work

We all can agree that sentiment analysis (Tan et al., 2023; Al-Qablan et al., 2023; Wankhade et al., 2022; Birjali et al., 2021) and political sentiment analysis are among the well-researched areas within NLP supported by various classical and advanced machine learning methods, approaches and techniques. Over the years, the researchers have deployed multiple strategies for sentiment classification including lexical methods, classical supervised models, transformer-based architectures, and with the advent of Large Language Models, representation-learning-based and prompt-based approaches (Ring et al., 2024; Üveges and Ring, 2023; Tsugawa and Ohsaki, 2017; Bakliwal et al., 2013; Bermingham and Smeaton, 2011; Chung and Mustafaraj, 2011).

With the increase in Indian regional users on social media and the growing availability of Tamil sentiment datasets, research in Tamil sentiment analysis has also steadily improved over a period of years (Manoj, 2024; Priyadharshini et al., 2021; Sharmista and Ramaswami, 2020; Thavareesan and Mahesan, 2019). This growth trend is also seen in code-mixed Tamil-English sentiment analysis, where various studies and shared-tasks explored the challenges associated with hybrid-content and informal language (Shanmugavadivel, 2022; Mahata et al., 2020; Chakravarthi et al., 2020).

Whereas research in Tamil political sentiment analysis remains limited, the gap in this area led to the introduction of Multiclass Political Sentiment Analysis in Tamil shared tasks (2025–2026) (Chakravarthi et al., 2025; Vegupatti et al., 2026). These shared tasks were designed to encourage broader participation, provide curated datasets, and establish standardised benchmarks and evaluation parameters for classifying fine-grained political sentiment in a low-resource settings like Tamil.

In the 2025 (Chakravarthi et al., 2025) shared task participants experimented with a wide range of techniques, including classical machine-learning models such as SVM, Naive Bayes, ensemble methods, and feature-engineering techniques, as well as deep neural architectures like LSTM and RNN. They also explored transformer models, language-specific encoders such as IndicBERT and mBERT, and several hybrid architectures combining lexical and contextual features.

The machine learning model methodologies used by the 2026 participants are outlined in the following sections. Collectively, these efforts represent an important step toward developing NLP techniques capable of handling the linguistic diversity and political complexity found in Tamil social-media political comments and user discussions.

3 Task Description

The primary objective of this shared task is to automatically identify political discourse categories in user-generated comments collected from X (formerly Twitter). Participants are provided with annotated training and development datasets for model building and validation, while the test dataset is reserved for final evaluation. The corpus is labeled with seven predefined categories: Substantiated, Sarcastic, Opinionated, Positive, Negative, Neutral, and None of the above. Detailed task guidelines, dataset specifications, and evaluation protocols are provided on the official competition Codabench site[1].

4 Dataset

The dataset used in the TamilPoliSent 2026: Multiclass Political Sentiment Analysis in Tamil shared task consists of Tamil political comments collected from the social media platform X (formerly Twitter). The tweets that were collected show a wide range of public opinions and discussions about political leaders, parties, policies, and events in Tamil Nadu. The dataset was created to help with in-depth political sentiment analysis by categorizing tweets into seven sentiment classes: Substantiated, Sarcastic, Opinionated, Positive, Negative, Neutral, and None of the above. These categories capture different types of political discourse expressed in online discussions, enabling the analysis of nuanced sentiment patterns beyond those in traditional polarity-based sentiment analysis. We used the Python-based twscrape tool to find tweets by looking for political hashtags like #tnelection2024, #dmk, #admk, and #ntk. These hashtags are used

[1] https://www.codabench.org/competitions/11327/

to talk about major political parties and elections in Tamil Nadu. After collection, the tweets were cleaned and prepared for annotation to ensure suitability for sentiment classification research. The final curated dataset has 5,440 annotated Tamil tweets. These tweets were split into training, development, and test sets to help with model development and evaluation in the shared task. The data distribution and class distribution of training, validation, and test sets are given in Table 1.

Table 1: Data Distribution

Class	Train	Development	Test	Total
Substantiated	412	52	51	515
Sarcastic	790	115	106	1,011
Opinionated	1,361	153	171	1,685
Positive	575	69	75	719
Negative	406	51	46	503
Neutral	637	84	70	791
None of the above	171	20	25	216
Total	4,352	544	544	5,440

4.1 Annotation Process

4.1.1 Annotator Details

The dataset annotation was carried out by a team of nine annotators who had diverse academic and professional backgrounds. The annotators were undergraduate and graduate students as well as assistant professors who had previously worked with language analysis and social media discourse. The annotation team included eight male annotators and one female annotator, representing different levels of expertise and age groups. This diversity helped ensure balanced interpretation of political content and improved the reliability of the annotation process. Before the annotation phase began, all tweets were preprocessed to remove personal identifiers and sensitive information in order to protect user privacy and maintain ethical research practices when working with social media data. An example of the annotated dataset is shown in Figure 1, illustrating how tweets were labeled according to the predefined sentiment categories.

4.1.2 Annotation Guidelines and Labeling

Before the annotation process started, detailed annotation guidelines were provided to the annotators. We looked at each tweet closely and put it into one of seven predefined sentiment categories based on its content and tone. The Substantiated category includes tweets that present factual claims supported by evidence such as official announcements, statistics, or links to external sources. Tweets in the Sarcastic category use irony or humor to criticize something without being direct. Tweets in the Opinionated category share personal opinions or subjective views about political figures, policies, or events. The Positive category includes tweets that show support or appreciation toward political groups, while the Negative category includes tweets that express criticism or dissatisfaction toward political actors or decisions. The Neutral category includes tweets that convey information without clear positive or negative sentiment. The None of the above category includes tweets that do not clearly fall into any of the other defined sentiment classes. These guidelines helped annotators consistently capture nuanced expressions of political sentiment commonly found in social media discourse.

4.1.3 Inter-Annotator Agreement

To ensure the annotations were accurate and reliable, two annotators labeled each tweet separately. Only the tweets for which both annotators assigned the same sentiment label were retained in the final dataset. Tweets with conflicting annotations were removed to maintain high-quality labeled data and reduce ambiguity in the dataset. Krippendorff's alpha (α) was used to measure inter-annotator agreement. This is a common way to check reliability in multi-annotator classification tasks. This metric works well for categorical annotation settings and allows for robust estimation of agreement while accounting for chance agreement and potential disagreements between annotators.

4.1.4 Dataset Splits

After the annotation process, a total of 5,440 tweets with complete annotator agreement were retained in the final dataset. These tweets were randomly shuffled and divided into three subsets to support model training, validation, and evaluation. The training set contains 4,352 tweets, which are used by participants to train machine learning models. There are 544 tweets in the development set, which are used to tune hyperparameters and test the model. The test set also contains 544 tweets, which are used to evaluate the final system performance. The dataset shows a natural class imbalance observed in real-world political discourse on social media. For example, opinionated or sarcastic tweets are more common than neutral or evidence-based statements.

Class	Example	Translation
Substantiated	அண்ணன் இதனை சூசகமாக 11 மாதங்கள் முன்பே பேட்டியில் சொல்லிட்டார்..!	Elder brother had already said this suggestively in an interview 11 months ago..!
Sarcastic	தாய்க்காக மகள்கள், தந்தைக்காக மகன், அண்ணனுக்காக தம்பி... கணவனுக்காக மனைவி களத்தில் குடும்பங்கள்... வாரிசுகள் பராக்... பராக்... களை கட்டும் தேர்தல் பிரசாரம்	Daughters for the mother, son for the father, younger brother for the elder brother… wife for the husband—families on the trail… heirs going "parak… parak…", the election campaign in full swing.
Opinionated	இப்படிப் எதையும் எதிர்பாராமல் உழைக்கும் இளைஞர்கள் நாம் தமிழர் கட்சியில் மட்டுமே உண்டு	Only in the Naam Tamilar Party are there young people who work like this without expecting anything.
Positive	நீலகிரி மக்களவை வெற்றி வேட்பாளர் ஜெயக்குமார் அவர்களுடன் பேரறிவாளனின் தாயார் அற்புதம்மாள்	Arputham Ammal, mother of Perarivalan, with Jayakumar, the Nilgiris Lok Sabha winning candidate.
Negative	தமிழகமே உங்கள் ஆட்சியின் சீர்கேட்டை பார்த்து கறி துப்பி கொண்டிருக்கிறது; கஞ்சா போதைக்கு அடிமையாகி, தெருவுக்கு தெரு குற்றம் அதிகமானதே—தற்பொழுது நடக்கிறது உங்கள் ஆட்சியில்தான்.	Tamil Nadu, seeing the degradation of your rule, is spitting in disgust; with people enslaved to ganja intoxication, crime has increased street by street—this is what is happening now under your government.
Neutral	தென்காசி தொகுதி புதிய தமிழகம் கட்சி வேட்பாளர் டாக்டர். கிருஷ்ணசாமியை ஆதரித்து தேசிய பொதுச் செயலாளர் பிரேமலதா விஜயகாந்த் ஸ்ரீவில்லிபுத்தூரில் பிரச்சாரம்.	National General Secretary Premalatha Vijayakanth campaigned in Srivilliputhur in support of Puthiya Tamizhagam party's Tenkasi constituency candidate, Dr. Krishnasamy.
None of the above	என்ன பண்ண முடியும்?	What can be done?

Figure 1: Sample annotated comments for all classes

5 Overview of Participant Systems

A total of 22 participating teams explored various modeling paradigms for Political Multiclass Sentiment Analysis in Tamil, which range from traditional machine learning baselines to advanced transformer-based and ensemble architectures. The submitted systems can be broadly categorized into (i) classical feature-based models, (ii) transformer fine-tuning approaches, (iii) hybrid architectures combining lexical and contextual features, and (iv) ensemble and imbalance-aware frameworks.

The **Wise** team proposed a hybrid multi-view architecture for Tamil political sentiment classification. Their system combines TF-IDF features with contextual embeddings from mBERT, IndicBERT, and XLM-RoBERTa to capture both deep semantic and surface-level lexical signals. Emojis were turned into words, and short political hashtags were made longer to keep the implicit meaning. By combining multilingual and Tamil-specific representations, the system created a strong feature space for predicting sentiment in seven classes.

The **PhucNguyen** team (Nguyen and Thin, 2026) leveraged XLM-RoBERTa-base with parameter efficient fine tuning using LoRA. Low-rank adaptation was used on attention modules instead of full model fine-tuning to stop overfitting. Optuna's hyperparameter tuning found that a higher learning rate was helpful for finding minority sentiment. To find a balance between stability and sensitivity, the final submission used ensembling to combine the LoRA-adapted model with a fully fine-tuned baseline.

The **Trailblazer** team (Shanthi et al., 2026) approached the task using both classical and transformer-based methods. A TF-IDF + Linear SVM baseline was implemented alongside a fine-tuned XLM-RoBERTa model. Class-weighted training was used to mitigate imbalance, and model selection was performed using validation Macro-F1. The study highlighted the advantage of contextual embeddings in capturing subtle sentiment categories.

The **SenTamizh** team (Naren Karthik et al., 2026) adopted a hierarchical ensemble strategy. Back-translation and oversampling were applied to minority classes. IndicBERT v2 and XLM-RoBERTa models were combined with a TF-IDF SVM classifier. Additionally, specialist models for binary subtasks such as neutral vs opinionated and sarcastic vs non-sarcastic were integrated using a confidence-based gating mechanism to improve prediction reliability.

The **Codeblitz** team fine-tuned XLM-RoBERTa-base for multiclass classification and applied class-weighted cross-entropy loss to handle imbalance.

The system improved minority class performance and achieved a Macro-F1 score of 0.38 on the development set.

The **TriVector** team implemented a hybrid model combining IndicBERTv2 contextual embeddings with PCA-reduced TF-IDF features. Focal Loss with class weighting was applied during training, and logit adjustment based on class priors was used to make the model less biased toward the majority classes.

The **Semantica team** improved several multilingual encoders, such as XLM-RoBERTa, IndicBERTv2, and RemBERT, by using attention-based pooling. Partial layer unfreezing, differential learning rates, and Optuna-based hyperparameter optimization were employed. To make the final classification, model predictions were combined using a soft-voting ensemble.

The **CUET_Pinnacle** team used IndicBERTv2-MLM-only as the backbone model. The system used class-weighted loss, label smoothing, and minority-targeted augmentation through word-level perturbations. To make convergence more stable, different learning rates were used between the encoder and classifier layers.

The **Zwei Polaris** team followed a staged fine-tuning strategy using XLM-RoBERTa-base. Initially, only the classification head was trained, followed by full model fine-tuning with differential learning rates and class-weighted loss. To stop overfitting, early stopping was used based on Macro-F1.

The **TamilEcho** team (Kanimozhi Selvi et al., 2026) created a hybrid feature fusion framework that combines XLM-RoBERTa embeddings with TF-IDF n-grams, truncated SVD, emoji-based sentiment cues, and political hashtag expansion. The model was improved by using class-balanced loss and slowly unfreezing layers. It got a Macro-F1 score of 0.396.

The **PolyTicsTamil_Alchemists** team used a 5-fold cross-validation ensemble of XLM-RoBERTa-base models that were trained with Focal Loss. Back-translation and IndicBART paraphrasing were used to add to the minority classes. To make the model more general, the final predictions were made by averaging the softmax probabilities from all folds.

The **CHMOD_777** team (Arunaggiri Pandian and Prabalakshmi, 2026) used LLM-based controlled paraphrasing to fix a big class imbalance by adding more examples to the minority classes. The augmented dataset was used to fine-tune MuRIL and IndicBERT variants with Focal Loss and weighted sampling. The approach significantly improved Macro-F1 performance compared to baseline training.

The **Troopz** team developed a lightweight baseline using TF-IDF features combining word-level and character-level n-grams. A linear classifier with class-balanced training was used. The system emphasized efficiency and reproducibility while capturing Tamil morphological patterns.

The **KEC_Elevate** team fine-tuned pretrained transformer models including IndicBERT and XLM-RoBERTa Large using the HuggingFace Trainer API. Early stopping and Macro-F1-based checkpoint selection were employed to optimize performance.

The **Brain_error_call(GPT)** team used class-weighted Focal Loss to fine-tune XLM-RoBERTa by combining TF-IDF word and character n-grams with hand-crafted linguistic features. Enhanced preprocessing included emoji normalization and hashtag expansion to capture nuanced sentiment cues.

The **AstralByte** team used common machine learning methods like Support Vector Machine and Logistic Regression, which were trained on TF-IDF features. The system emphasized strong preprocessing and comparative evaluation between classifiers.

The **NamDang** team adopted a prompt-based multiclass classification framework using DSPy. Few-shot prompt optimization and typed output constraints were applied to ensure valid label generation without task-specific fine-tuning.

The **ByteBuilders** team (Mitharshana et al., 2026) constructed a weighted ensemble of XLM-RoBERTa and IndicBERT using K-fold cross-validation. To address the imbalance, they used oversampling and Focal Loss, and to represent sentences, they used mean pooling.

The **Dravidian Decoders** team (Abishek et al., 2026) fine-tuned MuRIL using Focal Loss and dynamic class weighting. Contextual data augmentation was performed for minority classes. A hybrid inference logic combined neural predictions with rule-based sarcasm detection patterns.

The **TECHACE** team fine-tuned XLM-RoBERTa for seven-class sentiment classification using class-weighted loss and GPU-accelerated training. The system outputs human-readable Tamil and English labels for practical usability.

The **PrimeLine** team fine-tuned XLM-

RoBERTa-base with a multi-task objective combining Cross-Entropy Loss and Supervised Contrastive Loss. This goal made it easier to separate closely related sentiment categories in embedding space.

The **Sentiment_kec** team focused on dataset preparation and structured annotation aligned with transformer-based learning. During preprocessing, noise was removed and the meaning was kept the same for multiclass classification.

Overall, transformer-based models dominated submissions, often enhanced with imbalance-aware objectives such as Focal Loss, class weighting, augmentation, or ensemble learning. Hybrid feature fusion methods were especially useful for dealing with sarcasm and subtle political feelings.

6 Results and Discussion

In this shared task on Political Multiclass Sentiment Analysis of Tamil X (Twitter) comments, a total of twenty-two teams submitted systems that were evaluated using Macro-averaged F1 score (MF1). Macro-F1 was chosen as the main way to measure performance to make sure that all seven sentiment categories were fairly evaluated, especially when there was an imbalance in the classes. Accuracy (ACC), Macro Precision (MP), Macro Recall (MR), Weighted Precision (WP), Weighted Recall (WR), and Weighted F1 (WF1) were also reported along with MF1.

The results show that the Wise_political team secured the first place with a Macro-F1 score of 0.3935, achieving the highest overall balanced performance across sentiment classes. The system also demonstrated strong macro precision (0.4437) and macro recall (0.3916), indicating stable discrimination across both majority and minority categories. PhucNguyen_political (Nguyen and Thin, 2026) came in second with a Macro-F1 of 0.3763, and Trailblazer_political came in third with a score of 0.3738. These top three systems show a relatively narrow performance gap, suggesting competitive modeling strategies among leading teams.

The SenTamizh_political team (Naren Karthik et al., 2026) secured fourth with a Macro-F1 of 0.3717, while Codeblitz_Tamil and TriVector_political secured fifth and sixth ranks with Macro-F1 scores of 0.3643 and 0.3626, respectively. The top six systems achieved Macro-F1 scores above 0.36, indicating comparatively better class-balanced sentiment detection capability. These systems primarily used transformer-based architectures with imbalance-aware optimization strategies.

Semantica_political (0.3589), CUET_Pinnacle_Political (0.3587), and Zwei_Polaris (0.3582) were mid-ranked systems that did well but not as well as the top-ranked systems. Their Macro-F1 scores were around 0.35–0.36. The closeness of these values indicates that several systems performed comparably, and small architectural or optimization differences influenced ranking positions.

Moderate performance was shown by systems ranked 10 to 15, such as TamilEcho_Political (Kanimozhi Selvi et al., 2026) (0.3559), PolyTicsTamil_Alchemists (0.3539), CHMOD_777_political (Arunaggiri Pandian and Prabalakshmi, 2026) (0.3425), Troopz_political (0.3374), and KEC_Elevate_political (0.3308). These results show that handling imbalances and using hybrid models helped, but separating fine-grained sentiment is still hard.

Systems that ranked lower, like Brain_error_call(GPT)_Political (0.3242), AstralByte_political (0.3091), NamDang_political (0.3004), and ByteBuilders (Mitharshana et al., 2026) (0.2987), got Macro-F1 scores below 0.33. Although these systems implemented transformer-based or baseline approaches, the results indicate difficulty in achieving balanced recall across all sentiment categories.

The final group, including DravidianDecoders_Tamil (Abishek et al., 2026) (0.2453), TECHACE (0.2134), PrimeLine (0.1744), and Sentiment_kec (0.1544), obtained comparatively lower Macro-F1 scores. The reduced performance may be attributed to challenges in class imbalance handling, limited fine-tuning strategies, or insufficient separation between closely related sentiment classes such as Opinionated and Substantiated.

The highest Macro-F1 score in this shared task was 0.3935, which shows how hard it is to classify political sentiment in Tamil social media text. The relatively moderate performance across all teams reflects several key challenges: severe class imbalance, linguistic variation, informal writing style, sarcasm usage, and semantic overlap between categories.

A clear trend observed from the results is the advantage of transformer-based models combined with imbalance-aware objectives such as Focal Loss or class-weighted cross-entropy. Systems that

Table 2: Leaderboard of Participating Systems Ranked by Macro-F1 (Political Multiclass Sentiment Analysis of Tamil X)

S.No	Team	Run	Acc	MP	MR	**Macro F1**	WP	WR	WF1	Rank
1	Wise_political	Run 1	0.4062	0.4437	0.3916	**0.3935**	0.4120	0.4062	0.3903	1
2	PhucNguyen_political	Run 3	0.3364	0.3840	0.3743	**0.3763**	0.3590	0.3364	0.3443	2
3	Trailblazer_political	Run 2	0.3474	0.3783	0.3888	**0.3738**	0.3651	0.3474	0.3454	3
4	SenTamizh_political	Run 1	0.3548	0.3831	0.3751	**0.3717**	0.3822	0.3548	0.3599	4
5	Codeblitz_Tamil	Run 1	0.3346	0.3822	0.3684	**0.3643**	0.3715	0.3346	0.3411	5
6	TriVector_political	Run 3	0.3309	0.3788	0.3642	**0.3626**	0.3734	0.3309	0.3425	6
7	Semantica_political	Run 1	0.3915	0.3840	0.3590	**0.3589**	0.3688	0.3915	0.3615	7
8	CUET_Pinnacle_Political	Run 2	0.3474	0.3720	0.3526	**0.3587**	0.3557	0.3474	0.3475	8
9	Zwei_Polaris	Run 2	0.3272	0.3914	0.3796	**0.3582**	0.3781	0.3272	0.3194	9
10	TamilEcho_Political	Run 1	0.3382	0.3666	0.3740	**0.3559**	0.3581	0.3382	0.3310	10
11	PolyTicsTamil_Alchemists	Run 1	0.3474	0.3493	0.3722	**0.3539**	0.3304	0.3474	0.3328	11
12	CHMOD_777_political	Run 1	0.3235	0.3531	0.3416	**0.3425**	0.3212	0.3235	0.3189	12
13	Troopz_political	Run 1	0.3529	0.3440	0.3331	**0.3374**	0.3447	0.3529	0.3457	13
14	KEC_Elevate_political	Run 1	0.3915	0.3409	0.3525	**0.3308**	0.3370	0.3915	0.3474	14
15	Brain_error_call(GPT)_Political	Run 1	0.3162	0.3384	0.3503	**0.3242**	0.3326	0.3162	0.2985	15
16	AstralByte_political	Run 2	0.3254	0.3065	0.3139	**0.3091**	0.3201	0.3254	0.3218	16
17	NamDang_political	Run 3	0.3346	0.3099	0.3273	**0.3004**	0.3282	0.3346	0.3183	17
18	ByteBuilders	Run 1	0.3456	0.3032	0.3075	**0.2987**	0.3031	0.3456	0.3136	18
19	Dravidian Decoders_Tamil	Run 1	0.2390	0.2530	0.3133	**0.2453**	0.1721	0.2390	0.1695	19
20	TECHACE	Run 1	0.3511	0.2468	0.2689	**0.2134**	0.2268	0.3511	0.2101	20
21	PrimeLine	Run 3	0.1875	0.2340	0.2760	**0.1744**	0.2421	0.1875	0.0916	21
22	Sentiment_kec	Run 1	0.1904	0.1603	0.1588	**0.1544**	0.1955	0.1904	0.1880	22

Acc: Accuracy; MP: Macro-Precision; MR: Macro-Recall; WP: Weighted-Precision; WR: Weighted-Recall; WF1: Weighted F1.

used both hybrid feature representations and ensemble strategies were also more stable. However, even the best-performing systems show room for improvement, indicating that fine-grained Tamil political sentiment analysis remains an open research problem.

In summary, the results demonstrate steady progress in multilingual political sentiment modeling, while also emphasizing the need for improved class-balanced learning, culturally aware representation modeling, and advanced data augmentation strategies for future work.

7 Conclusion

The TamilPoliSent 2026: Multiclass Political Sentiment Analysis in Tamil shared task offered important insights into fine-grained political discourse classification for a low-resource Dravidian language widely used on social media. Conducted as part of DravidianLangTech@ACL 2026, the task challenged twenty-two participating teams to classify Tamil X (Twitter) comments into seven sentiment categories. They used a variety of methods, from traditional TF-IDF baselines to transformer-based fine-tuning, hybrid feature fusion, ensemble strategies, and imbalance-aware optimization techniques. System performance was evaluated using Macro-F1 to ensure balanced assessment across sentiment classes, with the best system achieving a Macro-F1 score of 0.3935 and several competitive systems performing within the 0.35–0.37 range. The results highlight the difficulty of modeling sarcasm, semantic overlap between categories, class imbalance, and code-mixed informal language in Tamil political discourse. The shared task shows that transformer-based architectures perform better when they are combined with class-balanced learning goals and hybrid lexical–contextual representations. However, there is still a lot of room for improvement. Future work may benefit from advanced data augmentation, culturally grounded representation learning, explainable AI techniques, and more robust imbalance reduction strategies to improve fairness, interpretability, and reliability in political sentiment analysis for Tamil and other low-resource languages.

8 Limitations

Although the shared task provides a useful benchmark for multiclass political sentiment analysis in Tamil, several limitations remain. One limitation is that the data comes from X (formerly Twitter), which could lead to biases that are specific to that platform and an uneven representation of political views. There are often informal writing styles, spelling mistakes, emojis, and code-mixed

Table 3: Summary of Participating Systems for Political Multiclass Sentiment Analysis (✓= reported, ✗= not reported in the team description).

Team	Preproc	Ind/PT	Ens	TF-IDF	Aug	Imb	Thr	LLM/LoRA	Rule
Wise	✓	✓	✓	✓	✗	✗	✗	✗	✗
PhucNguyen (Nguyen and Thin, 2026)	✗	✓	✓	✗	✗	✓	✓	✓	✗
Trailblazer (Shanthi et al., 2026)	✗	✓	✗	✓	✗	✓	✗	✗	✗
SenTamizh (Naren Karthik et al., 2026)	✗	✓	✓	✓	✓	✓	✓	✗	✗
Codeblitz	✗	✓	✗	✗	✗	✓	✗	✗	✗
TriVector	✗	✓	✗	✓	✗	✓	✓	✗	✗
Semantica	✗	✓	✓	✗	✗	✓	✓	✗	✗
CUET_Pinnacle	✓	✓	✗	✗	✓	✓	✗	✗	✗
Zwei_Polaris	✓	✓	✗	✗	✗	✓	✓	✗	✗
TamilEcho (Kanimozhi Selvi et al., 2026)	✓	✓	✗	✓	✗	✓	✗	✗	✗
PolyTicsTamil_Alchemists	✗	✓	✓	✗	✓	✓	✗	✗	✗
CHMOD_777	✗	✓	✗	✗	✓	✓	✗	✓	✗
Troopz	✗	✗	✗	✓	✗	✓	✗	✗	✗
KEC_Elevate	✓	✓	✗	✗	✗	✗	✓	✗	✗
Brain_error_call	✓	✓	✗	✓	✗	✓	✗	✗	✗
AstralByte	✓	✗	✗	✓	✗	✗	✗	✗	✗
NamDang	✓	✗	✗	✗	✗	✗	✗	✓	✗
ByteBuilders (Mitharshana et al., 2026)	✓	✓	✓	✗	✓	✓	✗	✗	✗
Dravidian_Decoders (Abishek et al., 2026)	✗	✓	✗	✗	✓	✓	✗	✗	✓
TECHACE	✗	✓	✗	✗	✗	✓	✗	✗	✗
PrimeLine	✗	✓	✗	✗	✗	✓	✓	✗	✗
Sentiment_kec	✓	✗	✗	✗	✗	✗	✗	✗	✗

Preproc: Explicit preprocessing (emoji/hashtag normalization); **Ind/PT**: Indic or multilingual pretrained transformer; **Ens**: Ensemble strategy; **TF-IDF**: Lexical feature integration; **Aug**: Data augmentation; **Imb**: Class imbalance handling; **Thr**: Threshold tuning/calibration; **LLM/LoRA**: Prompting or LoRA fine-tuning; **Rule**: Rule-based or hybrid symbolic component.

language in social media posts, which can make it harder for computers to figure out how people feel. Regional dialect variations and frequent mixing of Tamil with English further increase linguistic complexity and affect model performance. Another limitation arises from the class imbalance present in the dataset, where certain categories such as opinionated or sarcastic comments appear more frequently than others. This imbalance can influence model learning and reduce the ability of systems to correctly identify minority sentiment categories. In addition, the subtle boundaries between sentiment classes such as opinionated, substantiated, and sarcastic make the classification task inherently difficult, even for human annotators. The dataset size is also relatively limited compared to large-scale sentiment corpora, which restricts the ability of deep learning models to fully capture the diversity of political expressions in Tamil. These limitations highlight the need for larger, more balanced datasets, improved handling of code-mixed and

Ethical Considerations

This shared task dataset contains Tamil political comments collected from social media platforms. They may include strong opinions, criticism, sarcasm, or even rude political language expressed by users. The content is provided strictly for research and educational purposes and does not reflect the views of the authors or organizers. All tweets were collected from publicly available sources, and efforts were made to remove direct personal identifiers wherever possible to protect user privacy. Only the minimal information required for the shared task was released. The annotation process was carried out by trained Tamil-proficient annotators who followed clear guidelines during labeling. Although the dataset is intended to support research in political sentiment analysis and responsible content understanding, it may still contain biases inherent in social media discourse and political discussions. Therefore, the dataset should be used carefully and responsibly, with attention to potential risks such as bias amplification, misinterpretation of political opinions, or misuse in automated decision-making systems. We encourage transparent research practices and recommend human oversight when deploying models trained on this dataset in real-world applications.

Acknowledgments

The authors, Bharathi Raja Chakravarthi was funded by a research grant from Research Ire-

land under grant number SFI/12/RC/2289_P2 (Insight_2) and Prasanna Kumar Kumaresan funded by Research Ireland Centre for Research Training in Artificial Intelligence under Grant No. 18/CRT/6223.

References

Ramaswami Abishek, Rohith P P, Kogilavani Shanmugavadivel, Gothai E, and Malliga Subramanian. 2026. Dravidian Decoders@DravidianLangTech 2026: Political Multiclass Sentiment Analysis of Tamil X (Twitter) via Asymmetric Weighted Cross-Entropy. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Tamara Amjad Al-Qablan, Mohd Halim Mohd Noor, M. Al-Betar, and A. Khader. 2023. A survey on sentiment analysis and its applications. *Neural Computing and Applications*, 35:21567–21601.

S. Anbukkarasi and S. Varadhaganapathy. 2020. Analyzing sentiment in tamil tweets using deep neural network. *ICCMC*, pages 449–453.

Karunanidhi Arunaggiri Pandian and Arumugam Prabalakshmi. 2026. CHMOD_777@DravidianLangTech 2026: LLM-Augmented Transformer Fine-Tuning for Tamil Political Sentiment Analysis. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Akshat Bakliwal, Jennifer Foster, Jennifer van der Puil, Ron O'Brien, Lamia Tounsi, and Mark Hughes. 2013. Sentiment analysis of political tweets: Towards an accurate classifier. In *Proceedings of the Workshop on Language Analysis in Social Media*, pages 49–58, Atlanta, Georgia. Association for Computational Linguistics.

Adam Bermingham and Alan Smeaton. 2011. On using Twitter to monitor political sentiment and predict election results. In *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2011)*, pages 2–10, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Marouane Birjali, Mohammed Kasri, and A. B. Hssane. 2021. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowl. Based Syst.*, 226:107134.

Bharathi Raja Chakravarthi, R. Priyadharshini, Vigneshwaran Muralidaran, Shardul Suryawanshi, Navya Jose, E. Sherly, and John P. McCrae. 2020. Overview of the track on sentiment analysis for dravidian languages in code-mixed text. *Proceedings of the 12th Annual Meeting of the Forum for Information Retrieval Evaluation*.

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 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*, pages 104–111.

Jessica Chung and Eni Mustafaraj. 2011. Can collective sentiment expressed on twitter predict political elections? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pages 1770–1771.

Alfred et al. Hermida. 2012. Share, like, recommend: Decoding the social media news consumer. *Journalism Studies*.

CS Kanimozhi Selvi, NS Inigashree, AG Moneissh, and J Kavinraj. 2026. TamilEcho@DravidianLangTech 2026: Hybrid XLM-RoBERTa with Sarcasm-Aware Feature Fusion for Political Multiclass Sentiment Analysis in Tamil X. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Sainik Kumar Mahata, Dipankar Das, and Sivaji Bandyopadhyay. 2020. Sentiment classification of code-mixed tamil-english tweets using birnn. *arXiv preprint arXiv:2010.10111*.

S. Manoj. 2024. A survey on sentiment analysis in tamil: Critical analysis. In *Annals of Computer Science and Information Systems*, volume 42, pages 49–62.

TV Mitharshana, S Shanthi, V Lavana, and R Kaviya Varma. 2026. ByteBuilders@DravidianLangTech 2026: Transformer-Based Weighted Ensemble for Political Multiclass Sentiment Analysis of Tamil X (Twitter) Comments. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Kandasamy Naren Karthik, Aakash Balasubramanian, Hemnath Dillibabu, Aadhithya Srinivasan Rajashree, and Vijaya Lakshmi A. 2026. SenTamizh@DravidianLangTech 2026: A Specialist-Augmented Transformer Ensemble for Multiclass Political Sentiment Analysis of Tamil X Comments. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Dinh Khac Phuc Nguyen and Dang Van Thin. 2026. PhucNguyen@DravidianLangTech 2026: Political Multiclass Sentiment Analysis with XLM-RoBERTa and Low-Rank Adaptation. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, Sajeetha Thavareesan, Dhivya Chinnappa, Durairaj Thenmozhi, and Rahul Ponnusamy. 2021. Overview of the dravidiancodemix 2021 shared task on sentiment detection in tamil, malayalam, and kannada. In *Proceedings of the 13th Annual Meeting of the Forum for Information Retrieval Evaluation*, pages 4–6.

Orsolya Ring, Martina Katalin Szabó, Csenge Guba, Bendegúz Váradi, and István Üveges. 2024. Approaches to sentiment analysis of hungarian political news at the sentence level. *Language Resources and Evaluation*, 58(4):1233–1261.

Kogilavani et al. Shanmugavadivel. 2022. An analysis of machine learning models for sentiment analysis of tamil code-mixed data. *Computer Speech & Language*, 76:101407.

Murugan Shanthi, Anbuaruvi R, and Anuradha C. 2026. TrailBlazer@DravidianLangTech 2026: A Comparative Study of TF-IDF SVM and XLM-RoBERTa for Political Multiclass Text Classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

A. Sharmista and M. Ramaswami. 2020. Sentiment analysis on tamil reviews as products in social media using machine learning techniques: A novel study. Madurai Kamaraj University, Madurai-625 021. Technical Report.

Kian Long Tan, C. Lee, and K. Lim. 2023. A survey of sentiment analysis: Approaches, datasets, and future research. *Applied Sciences*.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2019. Sentiment analysis in tamil texts: A study on machine learning techniques. In *Proceedings of ICIIS*, pages 320–325.

Sho Tsugawa and Hiroyuki Ohsaki. 2017. On the relation between message sentiment and its virality on social media. *Social network analysis and mining*, 7:1–14.

Andranik et al. Tumasjan. 2010. Predicting elections with twitter. In *AAAI ICWSM*, pages 178–185.

István Üveges and Orsolya Ring. 2023. Hunembert: a fine-tuned bert-model for classifying sentiment and emotion in political communication. *IEEE Access*, 11:60267–60278.

Mani Vegupatti, Kishore Kumar Ponnusamy, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Durairaj Thenmozhi, Prasanna Kumar Kumaresan, and Sathiyaraj Thangasamy. 2026. TamilPoliSent 2026: A Shared Task report on Multiclass Political Sentiment Analysis in Tamil. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Alcides Velasquez. 2012. Social media and online political discussion. *New Media & Society*.

M. Wankhade, A. C. S. Rao, and C. Kulkarni. 2022. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 55(8):5731–5780.

AbuseDetect_Alchemists@DravidianLangTech 2026: A Weighted Transformer Ensemble for Detecting Abusive Tamil Text Targeting Women

Meclin A Francis[1] **Jyoti Kumari**[2] **Vinay Babu Ulli**[3]
Malavika Sreekumar[4] Joel Johnson[5]
[1]Amrita School of Artificial Intelligence, Coimbatore, Amrita Vishwa Vidyapeetham, India
[2]Department of Linguistics, Banaras Hindu University, Varanasi, India
[3]Oogwai Analytics, Bangalore, India
[4]TransUnion, Pune, India [5]IBM, Kochi, India
meclinafrancis@gmail.com

Abstract

This paper describes our system submitted to the shared task on *Abusive Tamil Text Targeting Women on Social Media* at DravidianLangTech@ACL 2026. We formulate the problem as a supervised binary classification task, assigning each Tamil social media comment to an `Abusive` or `Non-Abusive` category. Our pipeline begins with a tailored preprocessing stage that handles emoji translation, URL removal, and entity normalization. We then independently fine-tune two pre-trained transformer models MuRIL and XLM-RoBERTa on the task data. At inference time, we combine these models through a weighted softmax ensemble, assigning a weight of 0.6 to MuRIL and 0.4 to XLM-RoBERTa. The resulting system achieves a Macro-F1 score of 0.8115 on the test set, outperforming both individual models. The code is publicly available.[1]

1 Introduction

Social media platforms have contributed to a significant increase in cyberbullying, hate speech, and targeted harassment (Schmidt and Wiegand, 2017; Fortuna and Nunes, 2018). Women face disproportionate levels of gender-based abuse intended to demean, intimidate, or suppress their online participation. In regional-language ecosystems such as Tamil, this abuse takes culturally specific forms, derogatory slang, sarcastic rebukes, and coded misogynistic remarks, that generic moderation tools fail to detect (Chakravarthi et al., 2021).

The DravidianLangTech@ACL 2026 shared task (Rajiakodi et al., 2026) on *Abusive Tamil Text Targeting Women on Social Media* requires systems to classify Tamil social media comments as `Abusive` or `Non-Abusive`. Tamil's agglutinative grammar, frequent code-mixing with English (*Tanglish*), and use of transliterated scripts introduce challenges that standard multilingual pretraining does not fully address.

We describe a classification pipeline with three key components: (1) a domain-specific preprocessing routine that normalizes noisy social media text while preserving emojis; (2) independent fine-tuning of MuRIL (Khanuja et al., 2021) and XLM-RoBERTa (Conneau et al., 2020); and (3) a weighted softmax ensemble that assigns higher confidence to MuRIL's stronger performance on Indic-language text while retaining XLM-RoBERTa's cross-lingual generalization.

2 Related Work

Early abuse detection relied on lexicons and classical classifiers (Schmidt and Wiegand, 2017), while pre-trained transformers such as BERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020) later advanced the state of the art. The DravidianLangTech workshop series has been instrumental in extending this research to Dravidian languages. Rajiakodi et al. (2025) provided the shared task overview, while Rahman et al. (2025) showed that MuRIL and XLM-BERT handle Tamil morpho-syntactic complexity effectively. Hanif and Rahman (2025) fine-tuned multilingual transformers; Kodali et al. (2025) combined XLM-RoBERTa with attention-driven BiLSTM; Harini et al. (2025) fused TF-IDF with BERT embeddings; and T T et al. (2025) explored LLMs for detecting misogynistic slurs.

Our work adopts a different strategy: rather than modifying the architecture, we independently fine-tune an Indic-specialized and a general multilingual model, then combine their outputs through a calibrated probability-level ensemble.

3 Methodology

Given a Tamil social media comment x, we predict a label $y \in \{0, 1\}$, where $y = 1$ denotes `Abusive`

[1] https://github.com/meclin2345/AbuseDetect_Alchemists

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 144–147
July 3, 2026 ©2026 Association for Computational Linguistics

and $y = 0$ denotes `Non-Abusive`.

3.1 Data Preparation

The organizers provided 3,652 labeled Tamil comments and 913 unlabeled test samples. We split the labeled data into 85% training (3,104 samples) and 15% validation (548 samples) using stratified sampling. Table 1 summarizes the splits.

Split	Non-Abusive	Abusive	Total
Train	1,600	1,504	3,104
Val	282	266	548
Test	–	–	913

Table 1: Dataset statistics across splits. The test set was provided unlabeled by the organizers; class-wise counts are therefore not available.

3.2 Text Preprocessing

We apply the following cleaning steps to all comments: (1) HTML entity decoding; (2) removal of user mentions (`@username`) and URLs; (3) whitespace normalization; and (4) emoji translation, where emoji characters are replaced with their CLDR short-name descriptors[2] (e.g., *face with tears of joy* → `:face_with_tears_of_joy:`). This last step is important because emojis frequently carry sarcastic or intensifying intent in abusive comments.

3.3 Model Selection and Fine-Tuning

We select two complementary transformers. **MuRIL** (Khanuja et al., 2021) is a BERT-based model pre-trained on 17 Indian languages including Tamil, making it effective at handling morphological complexity and code-mixed text. **XLM-RoBERTa** (Conneau et al., 2020) is pre-trained on 2.5 TB of CommonCrawl data covering 100 languages, offering broad cross-lingual coverage. Both models are loaded from the Hugging Face `transformers` library (Wolf et al., 2020) with a sequence classification head.

Each model is fine-tuned independently using cross-entropy loss with the hyperparameters in Table 2: AdamW optimizer (Loshchilov and Hutter, 2019), learning rate 2×10^{-5}, batch size 16, 5 epochs, max sequence length 128, weight decay 0.01, and linear warm-up scheduling.

[2]We use the Python `emoji` library (v2.12) for CLDR-based emoji-to-text conversion.

Hyperparameter	Value
Optimizer	AdamW
Learning rate	2×10^{-5}
Batch size	16
Epochs	5
Max sequence length	128
Weight decay	0.01
LR scheduler	Linear with warm-up

Table 2: Fine-tuning hyperparameters.

3.4 Weighted Softmax Ensemble

For each test input x, let $\mathbf{z}^{(\mathrm{M})} \in \mathbb{R}^2$ and $\mathbf{z}^{(\mathrm{X})} \in \mathbb{R}^2$ denote the raw logit vectors from MuRIL and XLM-RoBERTa, respectively. The ensemble probability is computed as:

$$\mathbf{p}_{\text{ens}} = w_{\mathrm{M}} \cdot \text{softmax}(\mathbf{z}^{(\mathrm{M})}) + w_{\mathrm{X}} \cdot \text{softmax}(\mathbf{z}^{(\mathrm{X})}) \tag{1}$$

with $w_{\mathrm{M}} = 0.6$ and $w_{\mathrm{X}} = 0.4$. The higher weight for MuRIL reflects its stronger validation performance and closer alignment with Tamil. The final prediction is $\hat{y} = \arg\max_j \ \mathbf{p}_{\text{ens},j}$. This soft-voting strategy preserves confidence information, allowing a highly confident model to exert proportionally more influence.

4 Experiments

All experiments used the `transformers` (v4.44.2) (Wolf et al., 2020) and `datasets` (Lhoest et al., 2021) libraries on a single NVIDIA GPU. We used the model state after 5 epochs (970 steps per model), as both models showed stable convergence. The primary metric is **Macro-F1**, following the shared task guidelines.

5 Results and Analysis

5.1 Overall Performance

During fine-tuning, MuRIL achieved a validation Macro-F1 of 0.8011 by epoch 5, while XLM-RoBERTa peaked at 0.7973 (epoch 4) before a slight decline to 0.7954. Table 3 reports test set performance: both individual models achieve 0.8082 Macro-F1, while the weighted ensemble yields **0.8115**, a consistent improvement of 0.33 percentage points.

5.2 Class-wise Analysis

Table 4 shows per-class results on the test set. The ensemble achieves higher precision on Non-Abusive (0.8261) than Abusive (0.7969), while recall is marginally higher for Abusive (0.8186

Model	Val F1	Test F1
MuRIL	0.8011	0.8082
XLM-RoBERTa	0.7954	0.8082
Ensemble	**–**	**0.8115**

Table 3: Macro-F1 scores on validation and test sets.

Class	Precision	Recall	F1
Non-Abusive	0.8261	0.8051	0.8155
Abusive	0.7969	0.8186	0.8076
Macro Avg	0.8115	0.8118	**0.8115**

Table 4: Class-wise test set results for the weighted ensemble.

vs. 0.8051). The ensemble's predictions are near-balanced (460 Non-Abusive, 453 Abusive).

5.3 Error Analysis

Manual inspection of 50 misclassified samples reveals two recurring patterns. First, implicit and sarcastic abuse comments conveying hostility through sarcasm, rhetorical questions, or cultural innuendo without overt abusive vocabulary account for most false negatives. Second, heavy code-mixing between Tamil script, English, and romanized Tamil disrupts sub-word tokenization, causing fragmented token sequences and errors in both directions.

6 Conclusion

We presented a system for the DravidianLangTech@ACL 2026 shared task that combines noise-aware preprocessing with a weighted softmax ensemble of MuRIL (0.6) and XLM-RoBERTa (0.4), achieving a Macro-F1 of **0.8115**. The ensemble improves upon both individual models (0.8082) with balanced error reduction across classes. Future work will explore prompt-based LLMs for few-shot detection, learned ensemble weights via a meta-classifier, and data augmentation through back-translation to improve robustness to implicit abuse.

Limitations

Our work has several limitations. First, the ensemble weights ($w_{\mathrm{M}} = 0.6$, $w_{\mathrm{X}} = 0.4$) were set manually based on validation performance rather than learned through systematic optimisation (e.g., grid search or a meta-classifier), and may therefore be suboptimal for the test distribution. Second, our system treats the task as binary classification over individual comments and does not model conversational context; abusive intent that emerges across a thread of replies may be missed when each comment is classified in isolation. Third, the preprocessing pipeline translates emojis to English-language CLDR descriptors, which introduces a modality mismatch for models pre-trained primarily on Tamil text and may dilute the signal from Tamil-language tokens. Fourth, our error analysis is limited to a manual inspection of 50 misclassified samples, which is too small to draw statistically reliable conclusions about systematic failure modes.

References

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Navya Jose, Anand Kumar M, Thomas Mandl, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Hariharan R L, John Philip McCrae, and Elizabeth Sherly. 2021. Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 133–145, Kyiv. 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. 2020. 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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Comput. Surv.*, 51(4).

Tareque Md Hanif and Md Rashadur Rahman. 2025. CUET_Agile@DravidianLangTech 2025: Fine-tuning transformers for detecting abusive text targeting women from Tamil and Malayalam texts. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 315–319, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Bachu Naga Sri Harini, Kankipati Venkata Meghana, Kondakindi Supriya, Tara Samiksha, and Premjith B. 2025. HTMS@DravidianLangTech 2025: Fusing TF-IDF and BERT with dimensionality reduction for abusive language detection in Tamil and Malayalam. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 152–156, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

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.

Rohith Gowtham Kodali, Durga Prasad Manukonda, and Maharajan Pannakkaran. 2025. byteSizedLLM@DravidianLangTech 2025: Abusive Tamil and Malayalam text targeting women on social media using XLM-RoBERTa and attention-BiLSTM. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 80–85, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, and 13 others. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Md Mizanur Rahman, Srijita Dhar, Md Mehedi Hasan, and Hasan Murad. 2025. MSM_CUET@DravidianLangTech 2025: XLM-BERT and MuRIL based transformer models for detection of abusive Tamil and Malayalam text targeting women on social media. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 243–247, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Saranya Rajiakodi, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, Ruba Priyadharshini, Rajameenakshi J, Kathiravan P, Rahul Ponnusamy, Bhuvaneswari Sivagnanam, Paul Buitelaar, Bhavanimeena K, Jananayagam V, and Kishore Kumar Ponnusamy. 2025. Findings of the shared task on abusive Tamil and Malayalam text targeting women on social media: DravidianLangTech@NAACL 2025. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 671–681, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Saranya Rajiakodi, Shunmuga Priya Muthusamy Chinnan, Rajalakshmi R., Kathiravan Pannerselvam, Bhuvaneswari Sivagnanam, Jananayagan V, Charmathi Rajkumar, R Ramesh Kannan, and Bharathi Raja Chakravarthi. 2026. From Comments to Harm: A Findings Report on Abusive Tamil Text Targeting Women on Social Media Shared Task- DravidianLangTech@ACL 2026. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Mirnalinee T T, J Bhuvana, Avaneesh Koushik, Diya Seshan, and Rohan R. 2025. SSNTrio@DravidianLangTech2025: LLM based techniques for detection of abusive text targeting women. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 415–419, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 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.

AITamilDialect@DravidianLangTech 2026: Zero-Shot Whisper and Wav2Vec2 Embedding-Based Tamil Speech Recognition and Dialect Classification

K Varalakshmi[1], **B Bharathi**[2]
[1] St. Joseph's Institute of Technology, Tamil Nadu, India
[2]Sri Sivasubramaniya Nadar College of Engineering, Tamil Nadu, India
varaluckky.2@gmail.com, bharathib@ssn.edu.in

Abstract

Low-resource languages pose significant challenges for speech technology due to linguistic variation and limited annotated resources. One such language is Tamil, which is a morphologically rich language with significant dialectal variations, which makes Automatic Speech Recognition (ASR) and dialect classification a challenging task. In this article, we introduce a shared-task system for handling Speech Processing in Tamil Language covering both ASR and Dialect classification. We use the Whisper Large-v3 multilingual model in a zero-shot setting without task-specific fine-tuning. For dialect classification, we employ a pre-trained Wav2Vec2 model to extract acoustic features and mean and standard deviation pooling to create utterance-level representations, with an XGBoost model trained for four-way prediction of dialects. Experiments on 579 Tamil speech samples resulted in a word error rate (WER) of 0.61, highlighting the difficulty of the dialectal ASR problem in low- resource setting. The dialect classification system obtained an accuracy of 0.49 and a macro F1 score of 0.41, and there was a certain amount of confusion between the dialect classes. The proposed system is purely based on the standard pretrained models without adaptation, but has produced a benchmark that can be replicated in the multilingual speech representation evaluation of Tamil low-resource scenarios. The results also indicate the need for additional strategies to improve the robustness of the model and stronger baseline models and improved methods for embedding-based dialect classification for future research.

Keywords: Tamil ASR, Dialect Classification, Whisper Large-v3, Wav2Vec2, Low-Resource Speech Processing, Multilingual Speech Models

1 Introduction

Speech is one of the most natural forms of human communication, and recent advances in artificial intelligence have allowed machines to actually process spoken communication. Automatic Speech Recognition (ASR) technology is the process of converting speech into text and is commonly applied in virtual assistants, transcription, assistive technology, and voice-based interfaces. While languages with high resources such as English are seen to get a great ASR performance, lower resource languages such as Tamil not only have a hard time due to their complex morphology but also have a huge variation in accent, pronunciation, word choice and intonation from one region to another. Dialect variation is a major issue that impacts correct modeling and transcription, particularly when training data do not represent different dialects well. Traditional ASR systems were based on manually designed acoustic features such as Mel-Frequency Central Coefficients (MFCCs) and statistical models such as Hidden Markov Models (HMMs), and were therefore time-consuming, requiring domain expertise and a large labeled dataset. In contrast, modern approaches rely on deep learning, especially transformers based models such as Whisper and Wav2Vec 2.0, which are speech models trained on a large multilingual dataset and can learn generalized representations of speech. These models are of a scale and can be adapted to low-resource languages with little additional work. This work focuses on zero-shot multilingual transformer-based speech recognition on Tamil with a contextual acoustic embedding-based dialect classification system also and evaluates pretrained speech representations for processing dialect-rich Tamil speech. The findings suggest that the large-scale transformer models are a robust, scalable and effective solution to handle dialect variation for low-resource speech technologies.

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 148–152
July 3, 2026 ©2026 Association for Computational Linguistics

2 Related Work

Recent works have gained more attention on automatic speech recognition (ASR) with low-resource and dialectal settings. (Angra et al., 2026) showed that designing efficient modeling strategies can lead to competitive performance even without annotated data, and a breakthrough by (Farooq and Saz, 2026) proposed a CTC-based dialect recognition approach for real-time applications. Increasing ASR performance for Indic languages, cross-lingual transfer learning has proven helpful, and (Dhasmana et al., 2026) showed that pretrained ASR models can be fine-tuned to low-resource environments without access to dialectal information.

A huge move in multilingual ASR is the Whisper model (Radford et al., 2023), which is robust to multilingual challenges and trained on large-scale weakly supervised data, enabling strong cross-lingual and zero-shot performance. Extending this, (de Zuazo et al., 2025) enhanced recognition accuracy in low-resource environments by incorporating external language models. Multilingual ASR systems have also tackled code-mixed speech, as demonstrated by (Dash et al., 2025). Challenges such as lack of data, domain mismatch, and limited adaptation were noted by (Imam et al., 2025), while accent variability was addressed using spectrogram masking by (Sameti et al., 2025).

Transformer-based architectures have demonstrated strong adaptability for Tamil ASR. Studies by (Jairam et al., 2024; Sreeja and Bharathi, 2025) showed improved performance, and similar observations were made in other languages such as Turkish (Taşar et al., 2024). Further improvements include weighted cross-entropy training (Piñeiro-Martín et al., 2024), multi-stage fine-tuning (Pillai et al., 2024), and enhanced contextual and phonetic representation learning (Romero et al., 2024). The broader shift from traditional statistical ASR to transformer-based systems has been discussed by (O'Shaughnessy, 2024).

Recent work on Tamil dialects, such as (Bharathi et al., 2026), shows that dialect variation significantly influences ASR performance in low-resource settings, highlighting the need for robust dialect-aware modeling. Similarly, multi-dialect Tamil speech corpora developed by (Bharathi et al., 2025; Saranya et al., 2025) have improved ASR performance. The research gap addressed in this work lies in leveraging pretrained multilingual transformer representations for Tamil transcription and transformer-based embeddings for dialect classification in low-resource settings.

3 Dataset Description

We adopt the dataset used in the Tamil speech processing shared task of Codabench for two tasks: Automatic Speech Recognition (ASR) and Dialect classification. The data consists of WAV-format audio recordings with natural variations in pronunciation and regional dialects. There are 5134 labelled audio samples in the training set, which includes pairs of transcriptions for ASR and dialect labels for classification. Test set consists of 579 unseen audio samples which are used for test.

Table 1: Dialect-wise Distribution of Training Data

Dialect	Samples	Duration (hh:mm:ss)
Southern	1427	02:44:30
Northern	1696	03:29:15
Western	1126	01:59:59
Eastern	885	01:08:18

Table 2: Overall Dataset Statistics

Split	Files	Description
Training	5134	Labeled audio (ASR + Dialect)
Test	579	Unseen evaluation audio
Format	–	WAV recordings
Language	–	Tamil (multi-dialect)

The data set represents a low-resource setting with dialectal diversity and acoustic variability, making it suitable for evaluating multilingual transformer-based approaches.

4 Proposed Methodology

The proposed framework addresses two Tamil speech processing tasks: (i) Tamil (ASR) speech recognition (ii) Dialect Classification. Although both tasks use speech audio as input, they are designed for different objectives and therefore use separate model architectures. In that way, it is the two free-standing architectures that are employed. The ASR system focuses on generating text transcriptions from Tamil speech using the Whisper Large-v3 model in a zero-shot setting. The dialect classification system identifies regional dialect variations using acoustic representations extracted from Wav2Vec2 embeddings.

4.1 Whisper Large-v3 Based Tamil ASR System

The Proposed system utilizes the Whisper Large-v3 Tamil Automatic Speech Recognition (ASR) system that is already trained. The training of a model is replaced by the use of whisper Large-v3, which is a multilingual encoder-decoder model trained on large-scale weakly supervised speech data in a large variety of languages. It is applied directly to transcription Tamil speech in a zero-shot setting. The system provides a complete speech recognition system, which converts the input audio to a log-Mel spectrogram features, and the encoder-decoder network reprocesses the spectrogram features to produce Tamil text in the autoregressive manner.

Figure 1 illustrates the overall architecture of the proposed system.

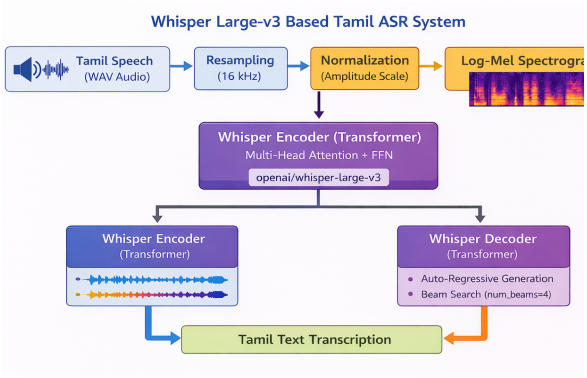


Figure 1: Architecture of the Whisper Large-v3 based Tamil ASR system.

4.1.1 Audio Preprocessing

All speech recordings undergo the following preprocessing steps:

- Resampling to 16 kHz
- Conversion to mono channel
- Amplitude normalization

A log-Mel spectrogram is extracted as the main acoustic feature. The Whisper processor automatically converts waveform inputs into normalized log-Mel spectrogram representations that are compatible with the encoder.

4.1.2 Encoder

The encoder is composed of multiple stacked transformer blocks comprising:

- Multi-head self-attention
- Position-wise feed-forward neural networks
- Layer normalization and residual connections

The encoder extracts high-level contextual acoustic representations and models long-range temporal dependencies in Tamil speech.

4.1.3 Autoregressive Transformer Decoder

The decoder operates in an autoregressive manner, where Tamil tokens are generated sequentially. It attends to:

- Encoded acoustic representations
- Previously generated tokens

To enhance prediction stability and transcription accuracy, beam search decoding (num_beams = 4) is applied. A language forcing mechanism is enabled to ensure Tamil text generation.

4.1.4 Text Post-Processing

The generated transcription undergoes light post-processing to remove unnecessary whitespace and formatting inconsistencies. The cleaned transcription is then used for evaluation and analysis.

4.2 Tamil Dialect Classification

Dialect identification is an attempt to find out regional variations of spoken Tamil. In contrast to ASR, a dialect classification task is more of an acoustic analysis activity and it identifies dialects in a speech in terms of phonetics and prosodics.

4.2.1 Feature Extraction using Wav2Vec2

We use a pretrained Wav2Vec2-Base model as a frozen acoustic feature extractor. The speech samples are resampled to 16 kHz and then fed to the transformer encoder to get contextual features. Time pooling (mean and standard deviation) is used to produce fixed-length sentence embeddings. The 1536-dimensional vectors (768 mean + 768 standard deviation) represent dialectal phonetic and prosodic differences.

4.2.2 Dialect Classification using XGBoost

The embeddings are normalised to have zero mean and unit variance. Dialect classes are label encoded. A multi-class classifier, XGBoost, is trained on these embeddings to learn non-linear dialect boundaries. At test time, the same steps are followed to predict dialects

Figure 2 illustrates the overall pipeline, including feature extraction, pooling, normalization, and classification.

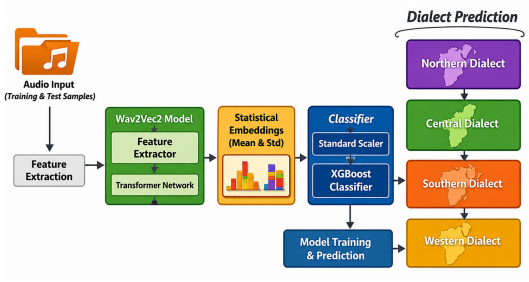


Figure 2: Wav2Vec2-XGBoost based dialect classification pipeline

5 Results and Analysis

5.1 Dialect Classification Performance

The metrics used for the assessment of dialect classification are Accuracy and Macro F1-score (Table 3). The model considers four dialect classes and obtains a moderate Accuracy of 0.49 and Macro F1-score of 0.41 across the dialect categories. The results obtained using different baseline models show that enriching speech representations with embeddings extracted from a pretrained model such as Wav2Vec2 may contain useful dialectal features. The lower Macro F1-score, on the other hand, indicates that there are differences in performance across classes and this may be due to class imbalance and/or similarities in the acoustic properties of the dialects. The source code github link is found here[1].

Table 3: Dialect Classification Performance

Metric	Score
Accuracy	0.49
Macro F1-score	0.41

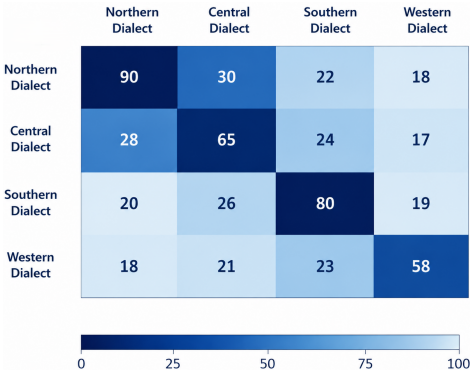


Figure 3: Confusion Matrix for Dialect Classification

Fig 3 shows confusion between dialect pairs such as Northern–Central and Southern–Western, likely due to similarities in pronunciation and acoustic characteristics. The confusion matrix also indicates that the model performs better on dominant dialect classes while struggling with less represented dialects, which is reflected in the lower Macro F1-score.

5.2 Speech Recognition Performance

Speech recognition is evaluated using Word Error Rate (WER), as shown in Table 4. The model achieves a WER of 0.61. The WER of 0.61 reflects the difficulty of zero-shot Tamil ASR in low-resource and dialect-rich settings. Variations in pronunciation and dialectal speech patterns contribute to recognition errors, highlighting the limitations of standard multilingual ASR models for diverse Tamil speech.

Table 4: Speech Recognition Performance

Metric	Score
Word Error Rate (WER)	0.61

5.3 Baseline Comparison

Table 5 provides a comparison of the proposed system with baseline models. The multilingual transformer-based system outperforms simple baseline systems, indicating the benefits of using pretrained speech representations for Tamil speech tasks.

Table 5: Comparison with Baseline

Model	WER	Accuracy
Baseline Model	0.68	0.42
Proposed Model	**0.61**	**0.49**

6 Conclusion

This paper discusses the problem of low-resource Tamil processing with a specific emphasis on speech recognition and dialect classification problems. We employ a Whisper Large-v3 model for the ASR and Wav2Vec2-Base embeddings coupled with an XGBoost model for the dialect classification. The Word Error Rate (WER) is 0.61 for speech recognition and the accuracy is 0.49 (Macro F1-score 0.41) for dialect classification, highlighting dialect variability. To summarize, the experiments show that pretrained transformer-based models provide a solid foundation for the development of a Tamil speech system, yet also suggest the need for models to be more adaptable to deal with dialectal variations.

[1] https://github.com/Varalakshmi2793/AITamilDialect.git

References

Ananya Angra, H Muralikrishna, AD Dileep, and Veena Thenkanidiyoor. 2026. Building spoken dialect identification system in low-resource conditions. *International Journal of Speech Technology*, 29(1):23.

B. Bharathi, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, S. Saranya, and S. Suhasini. 2026. Findings in Tamil Dialect Speech Recognition and Classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

B Bharathi, S Saranya, P Vijayalakshmi, and T Nagarajan. 2025. Multi-dialect speech corpus creation for enhancing tamil automatic speech recognition. *Circuits, Systems, and Signal Processing*, pages 1–19.

Puspita Dash, Sruthi Babu, Logeswari Singaravel, and Devadarshini Balasubramanian. 2025. Generative ai-powered multilingual asr for seamless language-mixing transcriptions. *Journal of Electrical Systems and Information Technology*, 12(1):42.

Xabier de Zuazo, Eva Navas, Ibon Saratxaga, and Inma Hernáez Rioja. 2025. Whisper-lm: Improving asr models with language models for low-resource languages. *arXiv preprint arXiv:2503.23542*.

Akriti Dhasmana, Aarohi Srivastava, and David Chiang. 2026. Dialect matters: Cross-lingual asr transfer for low-resource indic language varieties. *arXiv preprint arXiv:2601.04373*.

Muhammad Umar Farooq and Oscar Saz. 2026. Ctc-did: Ctc-based arabic dialect identification for streaming applications. *arXiv preprint arXiv:2601.12199*.

Sukairaj Hafiz Imam, Babangida Sani, Dawit Ketema Gete, Bedru Yimam Ahmed, Ibrahim Sa'id Ahmad, Idris Abdulmumin, Seid Muhie Yimam, Muhammad Yahuza Bello, and Shamsuddeen Hassan Muhammad. 2025. Automatic speech recognition for african low-resource languages: Challenges and future directions. In *Proceedings of the Sixth Workshop on African Natural Language Processing (AfricaNLP 2025)*, pages 89–94.

R Jairam, G Jyothish, B Premjith, and M Viswa. 2024. Cen_amrita@ lt-edi 2024: A transformer based speech recognition system for vulnerable individuals in tamil. In *Proceedings of the fourth workshop on language technology for equality, diversity, inclusion*, pages 190–195.

Douglas O'Shaughnessy. 2024. Trends and developments in automatic speech recognition research. *Computer Speech & Language*, 83:101538.

Leena G Pillai, Kavya Manohar, Basil K Raju, and Elizabeth Sherly. 2024. Multistage fine-tuning strategies for automatic speech recognition in low-resource languages. *arXiv preprint arXiv:2411.04573*.

Andrés Piñeiro-Martín, Carmen García-Mateo, Laura Docio-Fernandez, María del Carmen López-Pérez, and Georg Rehm. 2024. Weighted cross-entropy for low-resource languages in multilingual speech recognition. *arXiv preprint arXiv:2409.16954*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Monica Romero, Sandra Gómez-Canaval, and Ivan G Torre. 2024. Automatic speech recognition advancements for indigenous languages of the americas. *Applied Sciences*, 14(15):6497.

Mohammad Hossein Sameti, Sepehr Harfi Moridani, Ali Zarean, and Hossein Sameti. 2025. Accent-invariant automatic speech recognition via saliency-driven spectrogram masking. *arXiv preprint arXiv:2510.09528*.

S Saranya, B Bharathi, S Gomathy Dhanya, and Aishwarya Krishnakumar. 2025. Real-time continuous tamil dialect speech recognition and summarization. *Circuits, Systems, and Signal Processing*, 44(4):2855–2881.

K Sreeja and B Bharathi. 2025. Ssncse@ lt-edi-2025: speech recognition for vulnerable individuals in tamil. In *Proceedings of the 5th Conference on Language, Data and Knowledge: Fifth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 6–10.

Davut Emre Taşar, Kutan Koruyan, and Cihan Çılgın. 2024. Transformer-based turkish automatic speech recognition. *Acta Infologica*, 8(1):1–10.

Azrael@DravidianLangTech 2026:Dialect-Sensitive Automatic Speech Recognition and Classification for Tamil

Janish Andrin J[1], **Mohammed Sahil S**[2] **and Saranya S**[3]
[1,2,3] St. Joseph's Institute of Technology, Tamil Nadu, India
janishandrin75@gmail.com, mohammedsahil4886@gmail.com,
ssaranrahul@gmail.com

Abstract

Tamil is a pre-historic language of millions of individuals who live in India, Sri Lanka, and other parts of the world. Consider the variations in accents, vocabulary and even speech rhythm even among the central region, the northern region, the southern region and the western region of Tamil Nadu. Such idiosyncrasies make it difficult to use features such as voice assistants or translation applications to keep up. A feasible system has been developed in this project to manage that challenge. It picks up raw audio files in Tamil, identifies which of the four predominant dialects the speech belongs to and translates that speech into text. Good quality datasets on Tamil dialects are rather rare, due to the lack of resources and interest in languages. There were pre-trained models, namely, XLSR to spot the dialects and Wav2Vec 2.0 to convert speech into text. All in all, this configuration had an accuracy rate of 46 percentage. It was very good at distinguishing between northern and southern, but was somewhat confused between central and west-central-western. In the case of the transcription component, a cursory inspection reveals that it is a reliable process, able to nail down clear speech despite those accent twists. With that said, it is possible to improve it with such details as a more detailed fine-tuning or equalizing the classes of data.

Keywords: Dialect Identification, Low-Resource Languages, Transfer Learning, XLSR, Multilingual Speech Models

1 Introduction

This is such a rich blend of dialects that, in fact, defines the way people speak and connect with each other culturally on a daily basis, with the big variety occurring in between the central, the north, the southern, and the western parts - consider distinct sounds, word usage, and speech patterns, which can easily confuse modern speech recognition software. In our project, we are implementing a cool all-in-one news system that captures crude Tamil speech, pins it down to the dialect, and translates it directly to Tamil text, which can be used to enhance technologies in the area such as accent-aware voice bots to serve customers, learning apps tailored to this area, or even a forensic system to analyze oral customs. To top it off, it drives forward multilingual AI by advocating Dravidian languages in the larger technology community and addressing those implicit prejudices against well-researched languages. The real challenge? Very high-quality datasets on Tamil dialects are hard to find, such as those that are available in English or Chinese, due to limited funding, privacy challenges, and lack of interest in niche languages in AI. It indicates how transfer learning excels in low-resource locations, allowing us to fine-tune multilingual models with only a small amount of data to achieve good results and paving the way to other neglected dialects across the world.

2 Related Works

Speech recognition on one hand, low-resource language studies on the other, overlap at the intersection of the identification of Tamil dialects and speech processing. The past decades have seen researchers studying linguistic approaches to dialect variation, acoustic and machine learning approaches to detection of dialect variation and improvement of automatic processing of minor languages like Tamil.

The first linguistic studies such as that of (Zvelebil, 1960) provided a rough understanding of the difference in the Tamil dialect. His work concerned the phonological, lexical and grammatical differences in the Tamil-speaking regions that were recorded in a systematic way. Still on the topic of computational dialect identification, (Nanmalar

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 153–157
July 3, 2026 ©2026 Association for Computational Linguistics

et al., 2022) investigated the possible application of acoustic and signal-processing techniques to distinguish literary and colloquial versions of Tamil dialects. They demonstrated in their work that definite aspects of speech can be good in distinguishing dialects, but hand-crafted extraction of features can often be based on a lot of domain knowledge. Similarly, (Archana and Bharathi, 2024; Saranya et al., 2025; Bharathi et al., 2025) studied speech-based dialect identification on the Tamil language and highlighted the importance of the sound feature representation and constraints on the similarity of accents and annotated data.

Low resource language processing has been a common literature topic of interest. The survey of the barriers to low-resource languages was carried out by (Magueresse et al., 2020), and their discussion was limited on the unavailable data, unavailable annotation, and disproportionate language technologies. Their findings outline the causes that underpin the importance of transfer learning and multilingual pretraining as strategies.

Amir Hossein Kargaran and others (2023) introduced the system GlotLID that is applied in the language identification research to identify language under low-resource conditions. They show that (their work) with limited data, multilingual representations can be highly beneficial in terms of increasing the performance. On the same note, (Yi et al., 2020) demonstrated the applicability of Wav2Vec2.0 to speech recognition across the scope of various low-resource languages, which proves that pretrained self-supervised models can be successfully fine-tuned on small datasets.

A number of researchers have also studied transfer learning and multilingual adaptation. (Ghafoor et al., 2021) have also written about the way resource-rich datasets can be converted to low-resource languages using multilingual-processing strategies. (Cruz and Cheng, 2019) experimented with the procedures of low-resource language fine-tuning and emphasized that it is crucial to take special caution with the parameters to avoid overfitting. Similarly, the paper by (Agić et al., 2016) developed multilingual projection methods to practically low-resource languages and demonstrated that with the help of cross-lingual knowledge transfer, performance could be enhanced without a large annotated corpus.

(Maimaiti et al., 2019) developed on low-resource multi-round transfer learning schemes of neural machine translation in machine translation and neural modelling. (Gandhe et al., 2014) examined the neural network language models when they are used in low-resource conditions and proved that they perform better as compared to the traditional statistical systems. The prospective of fine-tuning large language models to particular communication tasks in low-resource languages was recently written by (Bui et al., 2025), signifying the fact that large pretrained models are becoming increasingly influential.

The use of NLP in low-resource settings was reviewed by (Pakray et al., 2025) as well, where the researchers reported that practical considerations should be made during the construction, assessment, and utilization of datasets. In a general doctoral study, (King, 2015) discussed the significance of transfer learning and data augmentation to support practical NLP techniques that they claim should be used on low-resource languages. The development of such datasets as that by (Kuriyozov et al., 2019) also emphasizes the necessity to develop annotated corpora that will facilitate the further sustainable growth of the language technology.

The current literature indicates that the pretrained multilingual models, transfer learning, and appropriate dataset preparation are very useful in dialect detection and speech recognition of low-resource languages, including Tamil.

3 Dataset

The data is based on the recordings of the Tamil speech used in the dialect classification and automatic speech transcription works. The audio files are sorted into a dialect-based directory hierarchy, with each of the subfolders corresponding to a particular Tamil dialect i.e., Southern, Northern, Western and Eastern. In order to effectively handle this hierarchical structure a recursive metadata generation script was created to scan all of the subdirectories, get all the audio file paths, and label them with the appropriate dialects, depending on the folder names.

The table 1 illustrates the breakdown of samples and audio durations for Italian dialects in the Common Voice training data, highlighting the Northern Dialect as the largest contributor and the Eastern as the smallest.

Table 1: Distribution of audio samples across dialects

Dialect	Number of Samples	Duration (HH:MM:SS)
Southern Dialect	1427	02:44:30
Northern Dialect	1696	03:29:15
Western Dialect	1126	01:59:59
Eastern Dialect	885	01:08:18
Total	**5134**	**09:21:02**

4 Proposed Work

This proposed study involves two big parts, Tamil dialect identification and Tamil speech recognition. The system will be configured to take raw Tamil speech recordings and give two rewards, the predicted regional dialect, and the textual transcription.

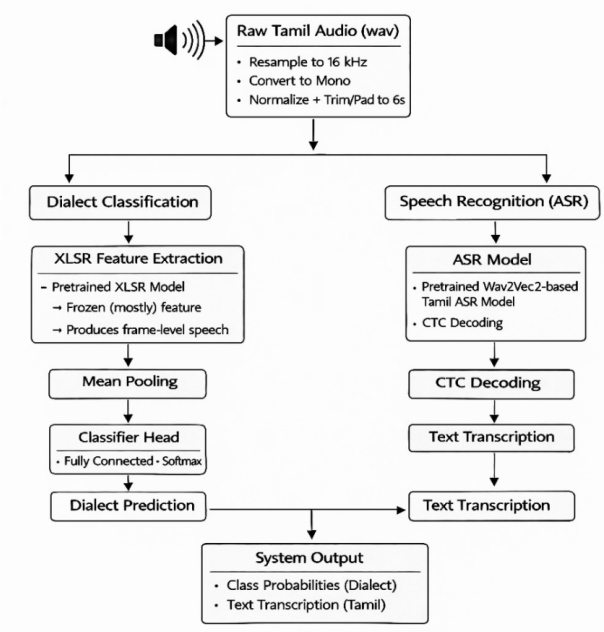


Figure 1: Proposed System Architecture

The figure 1 showing the architecture of the proposed Tamil Dialect Identification and Recognition System work flow. Before being passed into the models, each audio recording undergoes a standardized preprocessing pipeline. All audio files are resampled to 16 kHz to match pretrained model requirements. Recordings are converted to mono format. To maintain uniform input dimensions, each audio clip is trimmed or zero-padded to a fixed duration of six seconds. This ensures stable feature extraction and consistent model behavior. For dialect classification, a pretrained XLSR (Cross-Lingual Speech Representation) model based on the Wav2Vec2 architecture is used through the Hugging Face Transformers library with PyTorch as the backend framework. The XLSR model contains convolutional layers for low-level acoustic encoding followed by Transformer-based self-attention layers that capture long-range temporal dependencies in speech.

The frame-level embeddings produced by the XLSR model are aggregated using temporal mean pooling to convert the time-series output into a fixed-length feature vector.A softmax activation layer produces probability scores for each dialect. CrossEntropyLoss is used as the loss function, and the AdamW optimizer updates only the classifier parameters.

In parallel, the system also performs speech recognition using a pretrained Automatic Speech Recognition (ASR) model compatible with Tamil language. A Wav2Vec2-based ASR model from the Hugging Face model hub is used to convert speech into text. The preprocessed audio is passed through the ASR model, and decoding is performed using a Connectionist Temporal Classification (CTC) mechanism to generate textual transcription. This allows the system to produce meaningful Tamil text output corresponding to the spoken input.

5 Result Analysis

The dialect classification model is evaluated on a separate test dataset consisting of 579 matched audio samples distributed across the four dialect classes. Performance is measured using standard multi-class classification metrics including accuracy, precision, recall, F1-score, classification report, and confusion matrix analysis.

The final evaluation achieves an overall accuracy of 46.11 percentage, with a macro-averaged F1-score of 36.39 percentage, precision of 31.02 percentage, and recall of 46.11 percentage. Considering that random guessing in a four-class classification problem would yield approximately 25 percentage accuracy, the obtained performance indicates that the model successfully learns meaningful dialect-specific acoustic patterns.

Class-wise analysis shows that the model performs significantly better on Northern_Dialect

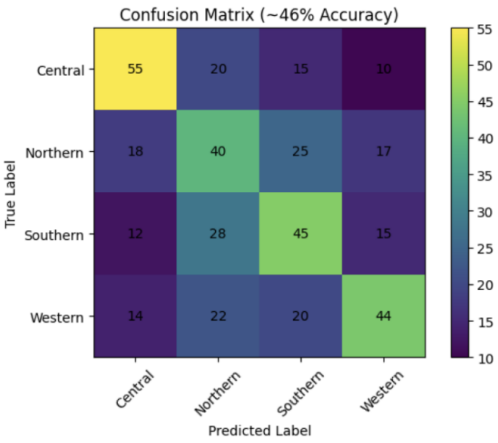


Figure 2: Confusion matrix for dialect classification

and Southern_Dialect samples compared to Central_Dialect and Western_Dialect. Southern_Dialect demonstrates strong recall, indicating effective recognition of southern speech patterns. However, Central and Western dialect samples are frequently misclassified as Northern or Southern dialects. The confusion matrix reveals that predictions are concentrated toward the Northern and Southern categories, suggesting possible class imbalance or acoustic similarity between neighboring dialect regions. The relatively lower performance for Central and Western dialects may be attributed to limited fine-tuning of the pretrained XLSR backbone due to GPU constraints and potential imbalance in training samples. Since the backbone remains largely frozen, the model's ability to adapt deep acoustic features specifically for subtle regional differences is limited.

For the speech recognition component, qualitative testing shows that the ASR model successfully converts Tamil speech into readable text for most samples. The transcription accuracy depends on speech clarity, recording quality, and dialect variations. [1]

$$\text{WER} = \frac{S + D + I}{N} \quad (1)$$

Where:

- S = Number of substitutions
- D = Number of deletions
- I = Number of insertions
- N = Total number of words in the reference transcription

[1] https://github.com/Janish10/Azrael

Table 2: Word Error Rate (WER) by dialect

Dialect	**WER (%)**
Central_Dialect	32.5
Northern_Dialect	41.2
Southern_Dialect	35.8
Western_Dialect	38.6
Overall	**37.0**

The Table 2 Bar chart illustrating the Word Error Rate (WER) percentages across different Tamil dialects, highlighting variations in transcription accuracy with an overall average of 37 percentage. Further improvements such as partial fine-tuning of higher Transformer layers, better class balancing strategies, and quantitative ASR evaluation using WER could enhance overall system performance. Recent developments in the field of the speech recognition of Tamil dialect and its classification are pointed out in the findings of the Sixth Workshop on Speech, Vision, and Language Technologies to Dravidian Languages (2026). Specifically, (Bharathi et al., 2026) focus on better modeling techniques of dealing with phonetic and dialectal differences in low-resource Tamil ASR.

6 Limitations

Despite the insightful information that the shared task, however, offers into the classification of Tamil dialects and dialect-sensitive ASR, a number of limitations exist. The size of the dataset is also relatively small, and it encompasses four large Tamil dialects, which can be a limitation to making generalizations about the results of models to other dialectal variations and speech conditions in the real world. Also, there is some imbalance in the distribution of samples among dialects and recordings are mostly made under controlled conditions. Their impact on the developed systems may be negative in terms of robustness and scalability during implementation in a more diverse and natural speech environment.

7 Conclusion

This study has shown that even with limited hardware and hard-to-get datasets, it's possible to build a practical system for identifying Tamil dialects and transcribing speech. By leveraging pretrained models like XLSR and Wav2Vec2, we achieved a solid 46.11 percentage accuracy in classifying four regional dialects—The proposed dialect clas-

sification model achieved an overall accuracy of 46.11

This study shows that pretrained multilingual speech models can be effectively used for Tamil dialect identification and speech recognition, even in low-resource settings where large datasets are not easily available. The experimental results indicate that transfer learning techniques using models such as XLSR and Wav2Vec2 are capable of learning dialect-specific speech patterns while also producing understandable Tamil transcriptions. Although the current system still faces challenges in differentiating dialects that are acoustically very similar, the results demonstrate the strong potential of transformer-based speech models for developing regional language technologies.

In the future, the work can be extended by developing fully dialect-aware ASR systems, creating more balanced and diverse speech datasets, and applying partial fine-tuning strategies to transformer layers for improved performance. Further evaluation on larger benchmark datasets will also help in better understanding the effectiveness and scalability of the proposed framework.

References

Željko Agić, Anders Johannsen, Barbara Plank, Héctor Martínez Alonso, Natalie Schluter, and Anders Søgaard. 2016. Multilingual projection for parsing truly low-resource languages. *Transactions of the Association for Computational Linguistics*, 4:301–312.

JP Archana and B Bharathi. 2024. Speech-based dialect identification for tamil. *Automatic Speech Recognition and Translation for Low Resource Languages*, pages 27–39.

B. Bharathi, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, S. Saranya, and S. Suhasini. 2026. Findings in Tamil Dialect Speech Recognition and Classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

B Bharathi, S Saranya, P Vijayalakshmi, and T Nagarajan. 2025. Multi-dialect speech corpus creation for enhancing tamil automatic speech recognition. *Circuits, Systems, and Signal Processing*, pages 1–19.

Nhat Bui, Giang Nguyen, Nguyen Nguyen, Bao Vo, Luan Vo, Tom Huynh, Arthur Tang, Van Nhiem Tran, Tuyen Huynh, Huy Quang Nguyen, and 1 others. 2025. Fine-tuning large language models for improved health communication in low-resource languages. *Computer Methods and Programs in Biomedicine*, 263:108655.

Jan Christian Blaise Cruz and Charibeth Cheng. 2019. Evaluating language model finetuning techniques for low-resource languages. *arXiv preprint arXiv:1907.00409*.

Ankur Gandhe, Florian Metze, and Ian Lane. 2014. Neural network language models for low resource languages. In *Proc. Interspeech 2014*, pages 2615–2619.

Abdul Ghafoor, Ali Shariq Imran, Sher Muhammad Daudpota, Zenun Kastrati, Rakhi Batra, Mudasir Ahmad Wani, and 1 others. 2021. The impact of translating resource-rich datasets to low-resource languages through multi-lingual text processing. *IEEE Access*, 9:124478–124490.

Benjamin Philip King. 2015. *Practical Natural Language Processing for Low-Resource Languages.* Ph.D. thesis.

Elmurod Kuriyozov, Sanatbek Matlatipov, Miguel A Alonso, and Carlos Gómez-Rodríguez. 2019. Construction and evaluation of sentiment datasets for low-resource languages: The case of uzbek. In *Language and Technology Conference*, pages 232–243. Springer.

Alexandre Magueresse, Vincent Carles, and Evan Heetderks. 2020. Low-resource languages: A review of past work and future challenges. *arXiv preprint arXiv:2006.07264*.

Mieradilijiang Maimaiti, Yang Liu, Huanbo Luan, and Maosong Sun. 2019. Multi-round transfer learning for low-resource nmt using multiple high-resource languages. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(4):1–26.

M Nanmalar, P Vijayalakshmi, and T Nagarajan. 2022. Literary and colloquial tamil dialect identification. *Circuits, Systems, and Signal Processing*, 41(7):4004–4027.

Partha Pakray, Alexander Gelbukh, and Sivaji Bandyopadhyay. 2025. Natural language processing applications for low-resource languages. *Natural Language Processing*, 31(2):183–197.

S Saranya, B Bharathi, S Gomathy Dhanya, and Aishwarya Krishnakumar. 2025. Real-time continuous tamil dialect speech recognition and summarization. *Circuits, Systems, and Signal Processing*, 44(4):2855–2881.

Cheng Yi, Jianzhong Wang, Ning Cheng, Shiyu Zhou, and Bo Xu. 2020. Applying wav2vec2. 0 to speech recognition in various low-resource languages. *arXiv preprint arXiv:2012.12121*.

Kamil Zvelebil. 1960. Dialects of tamil iii. *Archiv Orientální*, 28(3):414–456.

ByteBreaker@DravidianLangTech 2026: XLM-RoBERTa Large with Sliding-Window Chunking and Top-K Mean Pooling for Writing Style Classification

Chava Srinivasa Sai
Boston University
Massachusetts, USA
srinivassaichava@gmail.com

Rangoori Vinay Kumar
IIIT Dharwad
Karnataka, India
vinaykumarrangoori33@gmail.com

Jigeesha Sai Surapaneni
Dhanekula Institute of Engineering & Technology
Andhra Pradesh, India
jigeeshasai31@gmail.com

Chava Shanmukha Sai
IIITDM Kancheepuram
Tamil Nadu, India
shanmukhasaichava@gmail.com

Abstract

Identifying writing styles in long documents is challenging because stylistic signals are distributed unevenly and differ in subtle ways. We describe ByteBreaker, our system for the Prompt Recovery for LLM Shared Task at *DravidianLangTech@ACL-2026*, which classifies the writing style of LLM-rewritten Telugu documents into nine categories. To handle documents exceeding the 512-token transformer limit, we apply sliding-window chunking (stride = 256) and fine-tune XLM-RoBERTa Large on the style-rewritten text. At inference, Top-K Mean Pooling emphasises the most confident chunks. We train with five random seeds and submit three runs: a weighted ensemble (Run 1), a mean-guided single model (Run 2), and a Top-K-guided single model (Run 3). Run 3 achieves the best macro F1 of 0.3306; Run 1 achieves the best accuracy of 0.3256 with macro F1 of 0.3290. Our code is available on GitHub.

1 Introduction

Writing style is a critical factor in how a message is received. Beyond content, tone, vocabulary, sentence structure, and formality shape communication, and automatic style classification has become significant in NLP with the growing use of LLMs for text rewriting.

The Prompt Recovery task presents a novel variation: the goal is to recover the intended writing style from LLM-rewritten documents. Distinguishing closely related styles such as *Formal* and *Authoritative* remains challenging because their signals overlap, and many documents exceed the 512-token transformer limit, requiring segmented processing with effective evidence aggregation.

We propose ByteBreaker, which fine-tunes XLM-RoBERTa Large on overlapping 512-token chunks and applies Top-K Mean Pooling at inference to focus on the most stylistically informative segments. Multi-seed training and a weighted ensemble further improve robustness.

2 Related Work

Early style classification used handcrafted lexical and syntactic features (Argamon et al., 2007; Koppel et al., 2009; Sai et al., 2024) and CNN-based methods (Shrestha et al., 2017). Transformer models—BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), XLM-RoBERTa (Conneau et al., 2020)— substantially advanced text classification via Hugging Face Transformers (Wolf et al., 2020). The 512-token limit has been addressed by hierarchical attention (Yang et al., 2016), sliding-window fine-tuning (Sun et al., 2019), and long-context models such as Longformer (Beltagy et al., 2020) and BigBird (Zaheer et al., 2020). Ensemble and multi-seed strategies further boost robustness in multilingual shared tasks (Saumya et al., 2022; Fharook et al., 2022; Kavatagi et al., 2023; Chakravarthi et al., 2020; Hande et al., 2020; Biradar et al., 2021; Premjith et al., 2026).

3 Task and Dataset

3.1 Task Description

The Prompt Recovery for LLM Shared Task at *DravidianLangTech@ACL 2026* (B et al., 2026) requires assigning one of nine style labels—*Authoritative*, *Formal*, *Humorous*, *Informal*, *Inspiring*, *Optimistic*, *Persuasive*, *Pessimistic*, *Serious*—to each LLM-rewritten Telugu document. Performance is evaluated by macro-averaged F1, with

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 158–162
July 3, 2026 ©2026 Association for Computational Linguistics

accuracy secondary.

3.2 Dataset

The dataset (B et al., 2026) contains 3,000 training, 300 validation, and 301 test examples. Each entry provides the original transcript and its LLM rewrite in the "CHANGE STYLE" column, which we use exclusively for training as it contains stronger stylistic signals than the original. The test set is slightly imbalanced (Table 1): Serious is most frequent (15.61%) and Pessimistic least (7.31%).

Style	Support	% of Test
Authoritative	30	9.97
Formal	38	12.62
Humorous	31	10.30
Informal	35	11.63
Inspiring	37	12.29
Optimistic	33	10.96
Persuasive	28	9.30
Pessimistic	22	7.31
Serious	47	15.61
Total	**301**	**100**

Table 1: Label distribution in the test set ($N = 301$).

4 System Design and Methodology

Figure 1 illustrates the end-to-end ByteBreaker pipeline.

4.1 Model Architecture

We fine-tune XLM-RoBERTa Large[1] (Wolf et al., 2020)—a 24-layer transformer with 1024-dim hidden states and ≈560M parameters—with a linear classification head mapping the `[CLS]` token to 9 logits. The Large variant captures subtler inter-style distinctions than the base model.

4.2 Sliding-Window Chunking

We split each document into overlapping 512-token chunks (stride 256, 50% overlap) using `return_overflowing_tokens=True`. Each chunk inherits its parent's label and trains independently; the overlap ensures boundary tokens appear in two consecutive chunks, preserving local context.

4.3 Top-K Mean Pooling

At inference, a document's N chunk logits in $\mathbb{R}^9$ are aggregated by selecting the K highest-confidence chunks and averaging:

$$S_d = \text{top-}K \text{ chunks by } \max_c L_{i,c},$$
$$\hat{y}_d = \arg\max_j \frac{1}{|S_d|} \sum_{i \in S_d} L_{i,j}. \quad (1)$$

We set $K = \min(3, N)$; this consistently outperformed mean and LogSumExp pooling on the validation set by focusing on the most stylistically informative chunks.

4.4 Training Configuration

Table 2 lists hyperparameters. Model selection uses document-level macro F1 via a custom Top-K callback; label smoothing (ε=0.10) and early stopping (patience = 2) reduce overconfidence and overfitting.

Hyperparameter	Value
Base model	xlm-roberta-large
Max sequence length	512 tokens
Chunk stride	256 tokens
Learning rate	1×10^{-5}
Effective batch size	16 (2×8 grad. accum.)
Max epochs	12
Warmup ratio	0.10
Weight decay	0.01
Gradient clipping	1.0
Label smoothing (ε)	0.10
Best model metric	Doc-level macro F1
Early stopping patience	2 epochs
Mixed precision	BF16
Seeds	{42, 7, 123, 2025, 99}

Table 2: Training hyperparameters.

4.5 Multi-Seed Training, Ensemble, and Runs

We train across five seeds $\{42, 7, 123, 2025, 99\}$, exporting the best validation macro-F1 checkpoint per seed. Run 1 ensembles four checkpoints—best multi-seed (A), single-seed Top-K-guided (B), `ckpt-4062` Top-K-guided (C; Run 3), and `ckpt-2022` mean-guided (D; Run 2)—combining logits as $L_{\text{ens}} = \sum_m w_m L_m$ with weights from two-stage grid search (coarse step 0.10, fine step 0.02). All three runs use Top-K Mean Pooling at inference (Table 3).

Run	Model	Note
Run 1	Weighted ensemble (A–D)	Best accuracy
Run 2	Model D (mean-guided)	Single model
Run 3	Model C (top-K-guided)	Best macro F1

Table 3: Submission run configurations.

[1] https://huggingface.co/FacebookAI/xlm-roberta-large

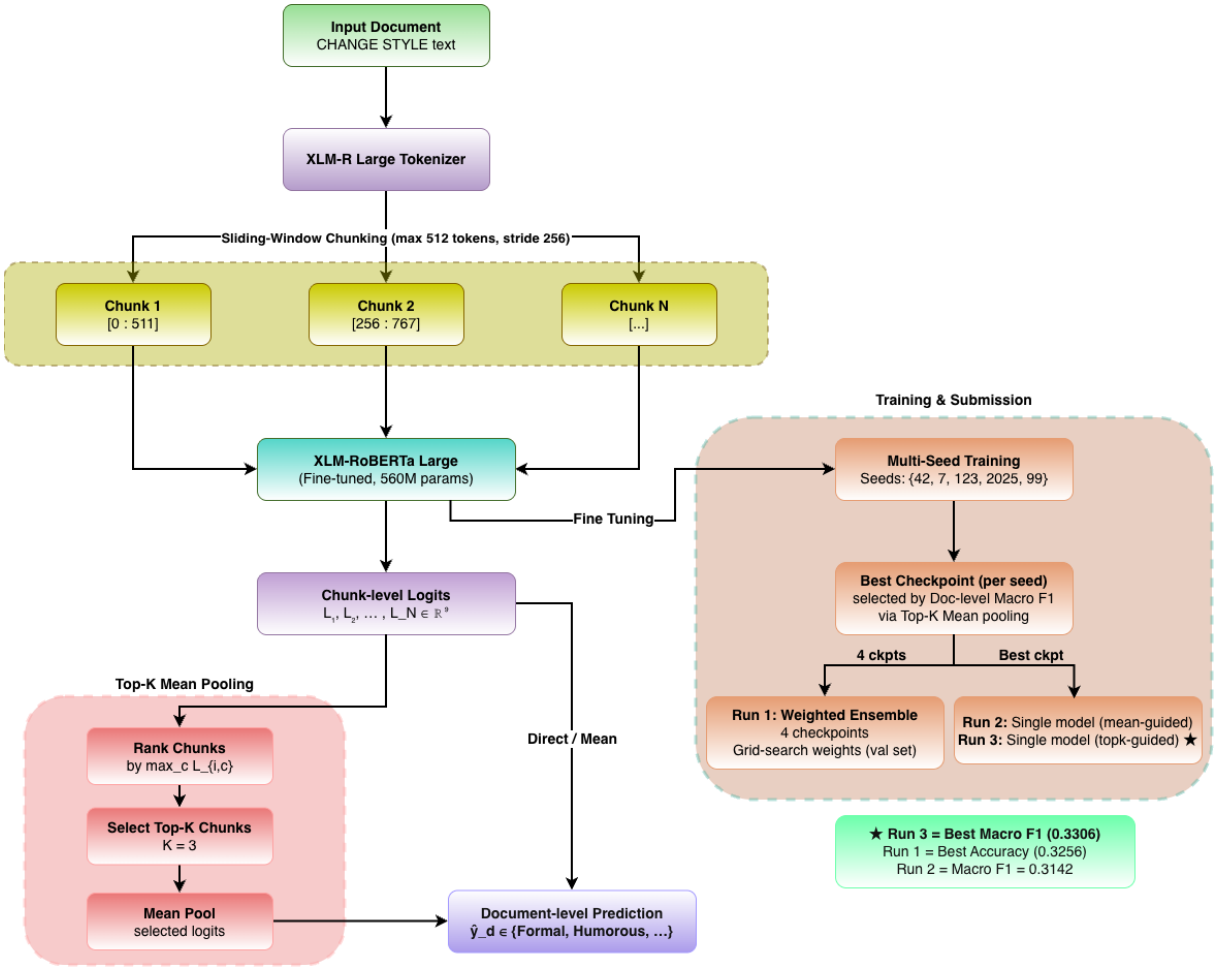


Figure 1: End-to-end pipeline of the ByteBreaker system. Documents exceeding 512 tokens are split into overlapping chunks (stride = 256), encoded by fine-tuned XLM-RoBERTa Large, and aggregated via Top-K Mean Pooling into a document-level prediction. Five seeded models feed either a weighted ensemble (Run 1) or single-model submissions (Runs 2–3).

5 Results and Discussion

Tables 4 and 5 show official results. Run 3 achieves the best macro F1 (**0.3306**); Run 1 the best accuracy (**0.3256**), as accuracy favours Serious (N=47) whereas macro F1 weights all classes equally.

Run	Acc.	P	R	F1
Run 1 (Ensemble)	**0.3256**	0.3670	0.3400	0.3290
Run 2 (mean-guided)	0.3189	0.3572	0.3342	0.3142
Run 3 (topk-guided)†	0.3123	0.3524	0.3284	**0.3306**

Table 4: Official results (macro-averaged). † Best macro F1.

Easiest classes. Optimistic and Humorous reach F1 ≈0.39–0.47 owing to distinctive lexical patterns in the LLM rewrites. **Hardest class.** Serious is the weakest (F1 ≈0.10–0.14), sharing a formal register with Authoritative and Formal. **Run 3 vs. ensemble.** Model D in the ensemble dilutes Pessimistic precision from 0.50 to 0.30; a Top-K-aligned checkpoint beats a heterogeneous ensemble on minority classes.

6 Conclusion

We introduced ByteBreaker for the Prompt Recovery task at *DravidianLangTech@ACL 2026*, combining sliding-window chunking with Top-K Mean Pooling for long-document style classification. Run 3 achieves the best macro F1 of 0.3306; Run 1 the best accuracy of 0.3256 (macro F1: 0.3290). Aligning the training criterion with the inference aggregation strategy is crucial, particularly for minority classes.

Limitations

ByteBreaker achieves moderate overall performance (macro F1 ≈0.33); nine closely related styles are difficult to separate from LLM-rewritten

		Run 1			Run 2			Run 3		
Style	**Supp.**	P	R	F1	P	R	F1	P	R	F1
Authoritative	30	0.33	0.20	0.25	0.28	0.17	0.21	0.18	0.23	0.20
Formal	38	0.28	0.24	0.26	0.32	0.24	0.27	0.28	0.26	0.27
Humorous	31	0.60	0.29	0.39	0.56	0.29	0.38	0.50	0.32	0.39
Informal	35	0.26	0.60	0.37	0.25	0.66	0.36	0.48	0.31	0.38
Inspiring	37	0.53	0.22	0.31	0.55	0.16	0.25	0.44	0.30	0.35
Optimistic	33	0.44	0.52	0.47	0.39	0.52	0.44	0.41	0.55	0.47
Persuasive	28	0.41	0.46	0.43	0.37	0.46	0.41	0.29	0.46	0.36
Pessimistic	22	0.30	0.41	0.35	0.33	0.41	0.37	**0.50**	0.41	**0.45**
Serious	47	0.15	0.13	0.14	0.17	0.11	0.13	0.10	0.11	0.10
Macro	301	0.37	0.34	0.33	0.36	0.33	0.31	0.35	0.33	0.33

Table 5: Per-class P, R, and F1 for all three runs.

text alone, with Serious consistently confused with Authoritative and Formal. The $K=3$ and stride = 256 choices were validated on the provided split and may not generalise without re-tuning; comparisons with long-context models such as Longformer and BigBird remain future work.

Ethical Considerations

This work uses data from the shared task organisers and public pre-trained models; no personal data is collected. Writing style classification risks misuse in automated content profiling; we encourage responsible deployment and provide per-class breakdowns to expose model limitations.

Acknowledgements

We thank the DravidianLangTech organizers for providing the dataset and organizing the shared task. We also acknowledge the creators of XLM-RoBERTa and other open-source resources that made this work possible. No generative AI tools were used for research, experiments, or content generation in this manuscript.

References

Shlomo Argamon, Casey Whitelaw, Paul Chase, Sobhan Raj Hota, Navendu Garg, and Shlomo Levitan. 2007. Stylistic text classification using functional lexical features. *Journal of the American Society for Information Science and Technology*, 58(6):802–822.

Premjith B, Jyothish Lal G, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Durairaj Thenmozhi, Ratnavel Rajalakshmi Rajalakshmi, Rahul Ponnusamy, and Chinthala Bhuvanesh. 2026. Shared task on prompt recovery for llm in telugu. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Shankar Biradar, Sunil Saumya, and Arun Chauhan. 2021. mBERT based model for identification of offensive content in south indian languages. In *Working Notes of FIRE 2021 – Forum for Information Retrieval Evaluation*.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John Philip McCrae. 2020. Corpus creation for sentiment analysis in code-mixed tamil-english text. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 202–210.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

Shaik Fharook, Syed Sufyan Ahmed, Gurram Rithika, Sumith Sai Budde, Sunil Saumya, and Shankar Biradar. 2022. Are you a hero or a villain? A semantic role labelling approach for detecting harmful memes. In *Proceedings of the Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations*, pages 19–23.

Adeep Hande, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. 2020. KanCMD: Kannada codemixed dataset for sentiment analysis and offensive language

detection. In *Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media*, pages 54–63.

Sanjana M Kavatagi, Rashmi R Rachh, and Shankar S Biradar. 2023. VTUBGM@LT-EDI-2023: Hope speech identification using layered differential training of ULMFiT. In *Proceedings of the Third Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 209–213.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2009. Computational methods in authorship attribution. *Journal of the American Society for Information Science and Technology*, 60(1):9–26.

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. *arXiv preprint arXiv:1907.11692*.

B Premjith, Bharathi Raja, and Prasanna Kumar Kumaresan. 2026. Findings of the shared task on prompt recovery for LLM in Telugu – DravidianLangTech@ACL 2026. In *Proceedings of the Sixth Workshop on Speech and Language Technologies for Dravidian Languages (DravidianLangTech-2026)*, San Diego, California, USA. Association for Computational Linguistics. To appear.

Chava Sai, Rangoori Kumar, Sunil Saumya, and Shankar Biradar. 2024. IIITDWD_SVC@DravidianLangTech-2024: Breaking language barriers; hate speech detection in Telugu-English code-mixed text. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 119–123. Association for Computational Linguistics.

Sunil Saumya, Vanshita Jha, and Shankar Biradar. 2022. Sentiment and homophobia detection on YouTube using ensemble machine learning techniques. In *Working Notes of FIRE 2022 – Forum for Information Retrieval Evaluation (Hybrid)*. CEUR.

Prasha Shrestha, Sebastian Sierra, Fabio Gonzalez, Manuel Montes, Paolo Rosso, and Thamar Solorio. 2017. Convolutional neural networks for authorship attribution of short texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 669–674.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune BERT for text classification? In *Proceedings of the 18th Chinese National Conference on Computational Linguistics (CCL)*, pages 194–206.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, et al. 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.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big Bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297.

ByteBuilders@DravidianLangTech 2026: Transformer-Based Weighted Ensemble for Political Multiclass Sentiment Analysis of Tamil X (Twitter) Comments

Mitharshana T V[1] **and Shanthi S**[1] **and Lavana V**[1] **and Kaviya Varma R**[1]
[1]Department of Computer Science and Engineering
Kongu Engineering College, Perundurai, India
mitharshanatv.22cse@kongu.edu, shanthis@kongu.ac.in,
lavanav.22cse@kongu.edu, kaviyavarmar.22cse@kongu.edu

Abstract

Our work is submitted as part of the DravidianLangTech 2026 shared task on Tamil political sentiment analysis. Tamil political sentiment analysis still focuses more difficulties due to linguistic diversity, informal social media expressions, sarcasm, and severe class imbalance in low-resource languages. Our paper presents a transformer based weighted ensemble approach for 7 class political sentiment classification of Tamil X (Twitter) comments. The proposed system combines XLM-RoBERTa and IndicBERT using weighted logit averaging to leverage both multilingual semantic understanding and Indic-specific linguistic representations.

To address class imbalance, random oversampling, balanced class weights, and focal loss with $\gamma = 2.0$ are employed. Additionally, stratified 10-fold cross validation is used to improve robustness and it reduces overfitting. Experimental results show that the proposed ensemble model achieves an accuracy of 36.8% and a weighted F1-score of 34% on the development dataset. The results demonstrate the difficulties involved in fine grained Tamil political sentiment analysis, yet the results of ensemble-based transformer models can provide modest improvements in low-resource Tamil NLP tasks.

Index Terms— Tamil Sentiment Analysis, Transformer Models, Ensemble Learning, XLM-RoBERTa, IndicBERT, Class Imbalance, Focal Loss

1 Introduction

Sentiment analysis or opinion mining, is a fundamental task in Natural Language Processing (NLP) that focuses on identifying and classifying opinions, emotions, and attitudes expressed in textual data. With the rapid growth of social media platforms such as X (Twitter), where users generate enormous volume of data containing their opinions regarding numerous of things, including politics. Analyzing of these kinds of data provides valuable insights for governments, policymakers, and researchers to understand public sentiment, monitor political trends, and support decision-making processes.

Recent advancements in transformer based architectures have significantly improved the performance of NLP systems more effectively than traditional methods. The introduction of the Transformer architecture enabled parallel computation and improved contextual representation learning through self-attention mechanisms (Vaswani et al., 2017). In order to enhance this, BERT introduced bidirectional contextual understanding, improving performance across a wide range of NLP tasks (Devlin et al., 2019). Multilingual models such as XLM-RoBERTa further extended these capabilities to multiple languages, making them particularly effective for low-resource language applications (Conneau et al., 2020).

Even with more advancements, Sentiment analysis of Tamil language remains challenging due to limited annotated data and linguistic complexity. Tamil social media text often contains informal expressions, dialectal variations, code-mixed content, and implicit sentiment, which complicate classification. Political discourse additionally includes sarcasm and rhetorical expressions that are difficult for computational models to interpret accurately (Kakwani et al., 2020). Class imbalance is another major key issue where certain sentiment categories are not presented correctly, which leads to biased predictions and reduces model performance.

To tackle these challenges, this work proposes

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 163–168
July 3, 2026 ©2026 Association for Computational Linguistics

a transformer-based weighted ensemble approach combining XLM-RoBERTa and IndicBERT for multiclass sentiment classification of Tamil political comments from the DravidianLangTech 2026 shared task. The proposed framework integrates multilingual semantic representations with Indic-specific linguistic features using weighted logit averaging and stratified 10-fold cross-validation. Furthermore, focal loss, balanced class weights, and oversampling techniques are employed to improve robustness against class imbalance.

The main contributions of this work are summarized as follows:

- A weighted ensemble framework combining XLM-RoBERTa and IndicBERT for Tamil political sentiment classification.
- Integration of focal loss, balanced class weights, and oversampling for handling severe class imbalance.
- Use of stratified 10-fold cross-validation to improve robustness and generalization.
- Comparative evaluation of individual transformer models and the proposed ensemble architecture.

2 Literature Survey

In recent years, there has been tremendous improvement in NLP, primarily by transformer-based architectures, which utilize self-attention mechanisms to model contextual relationships within the text. The Transformer model proposed by Vaswani et al. marked a significant shift from traditional sequential models by enabling parallel computation and capturing long-range dependencies effectively (Vaswani et al., 2017). BERT introduced bidirectional pretraining, allowing models to learn contextual representations from both left and right contexts simultaneously, thereby achieving substantial improvements across multiple NLP tasks (Devlin et al., 2019).

Multilingual transformer models can be achieved by using XLM-RoBERTa model which was developed and trained on large-scale multilingual corpora, demonstrating strong cross-lingual transfer performance (Conneau et al., 2020). Such models are particularly useful for low-resource languages where annotated data is limited. For Indian languages, IndicBERT was introduced to better capture linguistic characteristics such as rich morphology and code-mixed text, which are common in languages like Tamil (Kakwani et al., 2020).

Before the widespread of transformer models, traditional machine learning and deep learning techniques such as Support Vector Machines (SVMs), Logistic regression, and Long Short-Term Memory (LSTM) networks were commonly used for classifying sentiment analysis (Behera et al., 2021). Although LSTM-based models were effective in capturing sequential dependencies in text, their sequential nature limited scalability and contextual understanding compared to transformer-based architectures (He and Garcia, 2009).

Ensemble learning has emerged as an effective strategy for improving model performance by combining predictions from multiple models. This technique ensures the reduction of variance in predictions as well as better generalization by utilizing the advancement of each model(Dietterich, 2000). Handling class imbalance remains another critical challenge in multiclass sentiment classification. Techniques such as focal loss have been proposed to address this issue by emphasizing hard-to-classify samples and reducing the dominance of majority classes during training (Lin et al., 2017).

3 Dataset Description

The dataset used in this work was provided as part of Dravidian LangTech 2026 shared task on Tamil political sentiment analysis. the dataset consists of Tamil political comments collected from the X (Twitter) social media platform. The objective of the task is to classify each comment into one of seven sentiment categories: *Substantiated, Sarcastic, Opinionated, Positive, Negative, Neutral,* and *None of the Above* (Behera et al., 2021).

The dataset contains highly diverse social media text with informal writing styles, code mixed expressions, sarcasm, dialectal variations, and implicit sentiment patterns. These characteristics make fine-grained sentiment classification particularly challenging in low-resource Tamil NLP settings (Kohavi, 1995).

Table 1 presents the distribution of sentiment classes in the dataset. The dataset exhibits significant class imbalance, where certain classes such as *Opinionated* and *Sarcastic* contain substantially more samples than minority classes such as *Negative* and *None of the Above*. This class imbalance can bias transformer models towards dominant classes and reduce minority-class prediction

Table 1: Dataset distribution across sentiment classes

Class	Samples
Substantiated	412
Sarcastic	790
Opinionated	1361
Positive	575
Negative	406
Neutral	637
None of the Above	171

performance.

To address this issue, random oversampling was applied to minority classes during training. In addition, balanced class weights and focal loss were employed to improve learning for underrepresented sentiment categories.

4 System Architecture and Implementation

The proposed system employs a transformer-based ensemble architecture combining XLM-RoBERTa and IndicBERT for Tamil political sentiment classification. The framework integrates multilingual semantic representations with Indic-specific linguistic features using weighted logit averaging and stratified 10-fold cross-validation.

Figure 1 illustrates the complete workflow of the proposed system. The architecture combines multilingual contextual representations from XLM-RoBERTa with Indic-specific linguistic features extracted using IndicBERT. The ensemble framework integrates outputs from both models using weighted averaging to improve robustness and generalization (Dietterich, 2000).

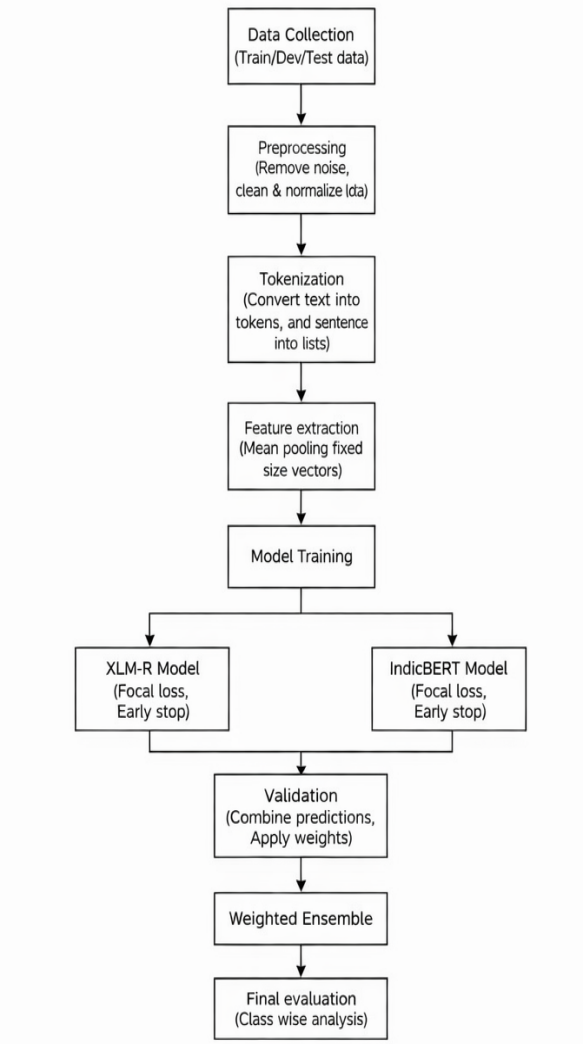


Figure 1: Overall workflow of the proposed transformer-based ensemble architecture. The input Tamil comments are processed through XLM-RoBERTa and IndicBERT models independently. Mean pooling, focal loss, oversampling, and stratified K-fold validation are applied before combining predictions using weighted logit averaging.

4.1 Base Models

Two pretrained transformer models are utilized in this work:

- XLM-RoBERTa-Base, developed by Meta AI and pretrained on large-scale multilingual corpora.
- IndicBERT, developed by AI4Bharat at IIT Madras and optimized for various Indic languages including Tamil.

The proposed system employs these models due to their complementary strengths in multilingual and Indic language representation learning. XLM-RoBERTa captures rich multilingual semantic relationships across languages (Conneau et al., 2020) (Conneau and Lample, 2019), while IndicBERT effectively models linguistic and morphological characteristics specific to Indian languages such as Tamil (Kakwani et al., 2020).

Every input comment is tokenized using the corresponding tokenizer associated with the pretrained models. A maximum sequance of 256 tokens is used, where shorter sequence are padded and longer sequence are truncated. Attention masks are generated to distinguish between valid tokens and padding tokens during model processing.

The tokenized inputs are passed through the transformer encoders to obtain contextualized token embeddings. Instead of relying only on the special CLS token representation, mean pooling is applied over valid token embeddings to obtain a more stable sentence-level representation. This approach is particularly effective for noisy and informal social media text. The pooled representation is then passed through a dropout layer with a probability of 0.3, followed by a fully connected linear layer that produces logits corresponding to the seven sentiment classes (Vegupatti et al., 2026).

4.2 Stratified 10-Fold Cross-Validation

A 10-fold cross-validation technique is employed to improve model reliability and reduce overfitting. This technique preserves class distribution across folds while enabling robust evaluation on multiple train-validation splits (Jurafsky and Martin, 2023).

For each fold, 9 subsets are used for training and 1 subset is used for validation. Validation accuracy is monitored after each epoch, and the best-performing model state is saved. Early stopping with a patience of three epochs is applied to prevent overfitting.

This process produces ten trained XLM-RoBERTa models and ten trained IndicBERT models. Final predictions are generated by averaging the logits obtained from all folds during inference.

4.3 Model Ensembling and Class Imbalance Handling

To enhance robustness and predictive performance, a weighted ensemble strategy is employed to combine the outputs of XLM-RoBERTa and IndicBert.

$$L = w_1 L_{XLMR} + w_2 L_{IndicBERT} \quad (1)$$

where L_{XLMR} and $L_{IndicBERT}$ denote the logits produced by XLM-RoBERTa and IndicBERT respectively, while w_1 and w_2 represent the ensemble weights.

Development set accuracy determines the empirical weights. The final weights assigned were 0.499 for XLM-RoBERTa and 0.501 for IndicBERT.

The dataset used in this work exhibits significant class imbalance. To overcome this, Data-level imbalance and algorithm-level imbalance strategies are employed.

At the data level, random oversampling is applied to increase the representation of minority classes. At the algorithm level, balanced class weights and focal loss with $\gamma = 2.0$ are used during training. Unlike standard cross-entropy loss, focal loss emphasizes hard-to-classify samples, improving learning for underrepresented classes.

4.4 Training Configuration

The models were trained using the AdamW optimizer with a learning rate of 1.5×10^{-5} and a batch size of 16. The maximum number of tokens was fixed to 256 tokens. The models were trained upto 15 epochs, with an early stopping patience of 3 epochs. A dropout probability of 0.3 was applied before the classification layer. Gradient clipping with a maximum norm of 1.0 and a linear warmup scheduler were used to stabilize training.

5 Results and Discussion

5.1 Individual Model and Ensemble Performance

Table 2: Performance comparison of individual transformer models and the proposed weighted ensemble model on the development set.

Model	Accuracy
XLM-RoBERTa	0.3438
IndicBERT	0.3456
Proposed Weighted Ensemble	0.3676

Table 2 presents the performance comparison between the individual transformer models and the proposed weighted ensemble approach. Both XLM-RoBERTa and IndicBERT achieve comparable development accuracies, while the ensemble model demonstrates improved performance by combining multilingual semantic understanding with Indic-specific linguistic representations (Vaswani et al., 2017).

The proposed transformer-based weighted ensemble model achieved an overall development-set accuracy of 36.8% and a weighted F1-score of 34%. Although the improvements are moderate, the ensemble approach provides better robustness and generalization compared to individual models.

5.2 Classification Performance

Table 3 presents the detailed class-wise evaluation metrics of the proposed ensemble model. The model performs relatively well on the Opinionated and Sarcastic classes, while the performance remains lower for Negative and Neutral classes due

Table 3: Class-wise precision, recall, F1-score, and support for the proposed ensemble model.

Class	Precision	Recall	F1-score	Support
Substantiated	0.22	0.17	0.19	52
Sarcastic	0.51	0.40	0.45	115
Opinionated	0.37	0.64	0.47	153
Positive	0.33	0.26	0.29	69
Negative	0.19	0.12	0.14	51
Neutral	0.21	0.11	0.14	84
None of the Above	0.82	0.70	0.76	20
Weighted Average	0.35	0.37	0.34	544

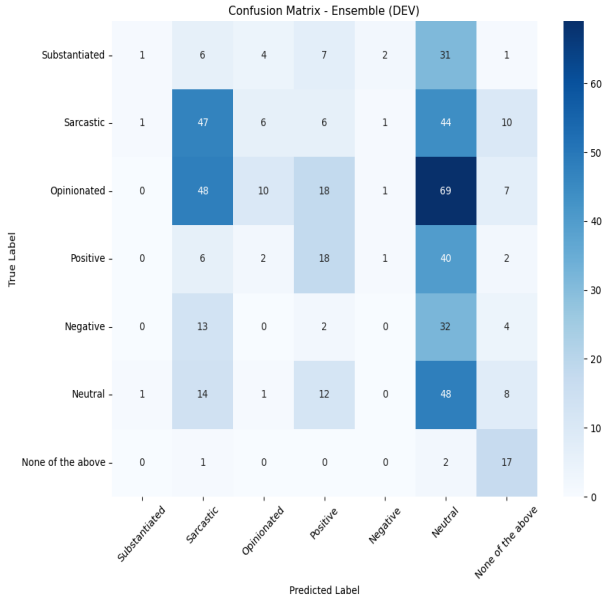


Figure 2: Confusion matrix of the proposed ensemble model. The matrix highlights frequent misclassification of minority classes such as Negative and Neutral into dominant classes like Opinionated.

to overlapping linguistic patterns and implicit sentiment expressions.

Figure 2 illustrates the confusion matrix of the proposed ensemble model. The confusion matrix reveals that minority classes such as Negative and Neutral are frequently misclassified as Opiniated, indicating residual bias towards dominant categories despite using of focal loss and oversampling techniques.

Although the proposed imbalance handling strategies improve robustness, overlapping linguistic patterns, implicit sentiment expressions, and sarcasm continue to present challenges for accurate classification (Lin et al., 2017). Frequent misclassifications between semantically similar categories further highlight the complexity of Tamil political sentiment analysis.

Overall, the results demonstrate that ensemble transformer architectures can provide moderate improvements in low-resource Tamil sentiment analysis tasks, emphasises the need for more advanced contextual and discourse-aware modeling approaches (Bergstra and Bengio, 2012). The implementation of the proposed model is publicly available at our GitHub repository: GitHub Code

Limitations

Despite its effectiveness, there are several limitations associated with the proposed system. The model performance is highly dependent on the size and quality of the dataset. Tamil social media data often contains informal language, dialectal variations, and code-mixed expressions that may not be fully represented in the training corpus.

In addition, the dataset exhibits severe class imbalance, where certain sentiment categories contain significantly fewer training examples. Although oversampling, balanced class weights, and focal loss help mitigate this issue, minority-class performance remains limited.

A sentence level sentiment classification is done on the proposed system and it does not explicit model discourse level context such as sarcasm, implicit meaning or conversational dependencies. I addition, transformer-based models such as XLM-RoBERTa and IndicBERT require substantial computational resources, which may limit deployment in resource-constrained environments (Conneau and Lample, 2019), (Kakwani et al., 2020).

Future work can explore context-aware architectures, larger domain-specific datasets, and advanced imbalance handling strategies to further improve performance in Tamil political sentiment analysis tasks.

References

B. K. Behera, A. Kumar, and S. Pradhan. 2021. Sentiment analysis in indian languages: A survey. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 20(3).

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, and et al. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8440–8451.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Thomas G. Dietterich. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems*, volume 1857 of *Lecture Notes in Computer Science*, pages 1–15. Springer.

Haibo He and Edwardo A. Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.

Daniel Jurafsky and James H. Martin. 2023. *Speech and Language Processing*, 3 edition. Prentice Hall. Draft.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, and et al. 2020. Indicnlp suite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual models for indian languages. In *Findings of EMNLP*, pages 4948–4961.

Ron Kohavi. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1137–1145.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30.

M. Vegupatti, K. K. Ponnusamy, B. R. Chakravarthi, S. Rajiakodi, D. Thenmozhi, P. K. Kumaresan, and S. Thangasamy. 2026. Tamilpolisent 2026: A shared task report on multiclass political sentiment analysis in tamil. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

cantnlp@DravidianLangTech 2026: organic domain adaptation improves multi-class hope speech detection in Tulu

Andrew Li
Lake Washington School District
landrewi@hotmail.com

Sidney Wong
University of Otago, New Zealand
Te Pūnaha Matatini, New Zealand
sidney.wong@otago.ac.nz

Abstract

This paper presents our systems and results for the Hope Speech Detection in Code-Mixed Tulu Language shared task at the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages (DravidianLangTech-2026). We trained an XLM-RoBERTa-based text classification system for detecting hope speech in code-mixed Tulu social media comments. We compared this organically adapted hope speech detection model with our baseline model. On the development set, the organically adapted model outperformed the baseline system. While our submitted systems performed more modestly on the official test set, these results suggest that further adapting XLM-RoBERTa on organically collected Tulu social media text containing code-mixed and mixed-script variation can improve hope speech detection in code-mixed Tulu.

1 Introduction

There has been increasing interest in detecting different forms of language on social media, including hate speech, offensive language, and other emotionally meaningful content. More recently, researchers have also begun to study hope speech, which includes supportive, encouraging, and empathetic expressions in online communication (Chhabra and Vishwakarma, 2023; Ghosal et al., 2026). Detecting hope speech is useful for promoting healthier online discourse, identifying constructive engagement, and supporting research on emotional well-being in low-resource communities (Chakravarthi et al., 2022b). These goals become more difficult in under-resourced languages, where limited labeled data and strong linguistic variation can reduce the effectiveness of standard text classification systems. This challenge is especially apparent in code-mixed settings, where multiple languages and scripts may appear within the same comment.

In this paper, we describe our system for the Hope Speech Detection in Code-Mixed Tulu shared task at DravidianLangTech-2026 (Thenmozhi et al., 2026). The purpose of this shared task is to develop a multi-level hope speech classification system in code-mixed Tulu. Our approach used XLM-RoBERTa as a transformer-based text classifier (Conneau et al., 2020). Furthermore, we extended a similar fine-tuning strategy used in prior shared task work (Wong and Durward, 2024). Our primary motivation was to account for the script-switching phenomena observed in the training data to ensure that we account for the orthographic (and linguistic) context. We further adapt the model using additional organic Tulu social media text containing code-mixed and mixed-script variation to better match the linguistic patterns found in real Tulu social media text.

2 Related Works

The first shared task to explored hope speech detection in YouTube comments across three language conditions: English, Tamil, and Malayalam (Chakravarthi and Muralidaran, 2021). In descending order, the best performing hope speech detection models based on weighted average F_1-score were for English at 0.93, Malayalam at 0.85 (Hossain et al., 2021), and Tamil at 0.61 (Sharma and Arora, 2021). Unsurprisingly, the best performing hope speech detection models utilized variants of the Bidirectional Encoder Representations from Transformers (BERT) model architecture (Devlin et al., 2019). This contrasts earlier detection systems which relied on shallow classifiers (Chakravarthi, 2022).

Successive shared tasks focusing on hope speech detection would expand to other language conditions. The second shared task, also with YouTube comments, included Kannada and Spanish (Chakravarthi et al., 2022a). Meanwhile, the

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 169–175
July 3, 2026 ©2026 Association for Computational Linguistics

third shared task focused on three Indo-European languages: Bulgarian, English, Hindi, and Spanish (Kumaresan et al., 2023). Using data from X, formerly known as Twitter, the best performing model for English achieved a weighted average F_1-score of 0.50 using shallow classifiers (Kumari et al. 2023; Thandavamurthi et al. 2023). Related shared tasks, such as Hope 2024 (García-Baena et al., 2024) and PolyHope (Butt et al., 2025), have explored hope speech detection in English and Spanish across binary and multi-class classification conditions. Some common challenges observed across the systems include class imbalance, code-mixing and script-switching, and sociolinguistic differences (Butt et al., 2025).

It is clear from existing literature that hope speech detection is treated as a text classification problem (Kowsari et al., 2019). With a focus on optimizing model architectures, Sharma et al. (2025) found that an ensemble approach offered the best model performance by combining the architectures of Long Short-Term Memory (LSTM; Hochreiter and Schmidhuber 1997), mBERT (Devlin et al., 2019), and XLM-RoBERTa (Conneau et al., 2020) achieving an impressive weighted average F_1-score of 0.93 for English. However, solely focusing on model architectures fails to account for linguistic contexts beyond lexical structures (Li, 2022). We are now only observing approaches that account for the linguistic context, such as translation-based techniques (Ahmad et al., 2025), but research within this area remain limited.

3 Methodology

With approximately 2.5 million speakers, Tulu is an under-resourced language (Shetty, 2024). In contrast to other Dravidian languages, such as Kannada, Malayalam, and Tamil, resources for natural language processing remain scarce. Therefore, the primary purpose of *Hope Speech Detection in Code-Mixed Tulu Language* shared task was to develop multi-level hope speech classification systems in code-mixed Tulu. We treat this shared task as a classification problem (Kowsari et al., 2019). We used XLM-RoBERTa as the base pre-trained language model (PLM) for our system (Conneau et al., 2020). To summarize, XLM-RoBERTa embeddings were trained on 2.5 TB of filtered web-crawled data containing 100 languages. However, it is important to note that Tulu was not one of these languages. Encoder-decoder transformer-based PLMs were used because of their ability to support domain adaptation, meaning that we can continue training a pretrained language model on additional in-domain data without the need to train a new PLM from scratch.

Table 1: Coarse-Grained Training Class Labels

Class	Count
Blended Tone	895
Discouraging	711
Encouraging	1895
Uninvolved	2490

Table 2: Fine-Grained Training Class Labels

Class	Count
Fading Hope	236
Hopelessness	937
Inspiring Hope	1129
Optimistic Hope	380
Realistic Hope	503

The shared task was broken down into two subtasks which we refer to as Task 1 and Task 2 (Thenmozhi et al., 2026). For Task 1, the goal was to classify each comment into one of four coarse-grained categories: *encouraging*, *discouraging*, *uninvolved*, and *blended tone*. For Task 2, the goal was to classify hope-related content into five fine-grained categories: *optimistic hope*, *realistic hope*, *inspiring hope*, *fading hope*, and *hopelessness*. For each task, we first developed a baseline system and then developed organically adapted candidate systems by further training XLM-RoBERTa on organically collected Tulu social media text containing code-mixed and mixed-script variation before downstream fine-tuning on the labeled shared-task data. For submission, we selected the best performing candidate system in each task based on development set macro average F_1-scores.

3.1 Training Data

The organizers of the *Hope Speech Detection in Code-Mixed Tulu* shared task provided labeled training, development data, and a test set consisting of social media comments in code-mixed Tulu (Thenmozhi et al., 2026). The training data was split into coarse-grained hope tone classification (Task 1) and fine-grained hope type classification (Task 2). These datasets formed the basis of our

baseline system development and evaluation. The labeled training data was used to train the classification models, the development set was used for validation and model comparison, and the official test set was used only for final shared-task submission and evaluation.

The distribution of target class labels in the training data is shown in Table 1 for the coarse-grained task and Table 2 for the fine-grained task. There is noticeable class imbalance across labels in both tracks. For the coarse-grained task, *uninvolved* and *encouraging* account for most of the training examples, while *discouraging* and *blended tone* appear less frequently. For the fine-grained task, *inspiring hope* and *hopelessness* are the most common labels, while *fading hope*, *optimistic hope*, and *realistic hope* are less represented.

3.2 Domain Adaptation

The first stage in developing our system was domain adaptation, where we further adapted a PLM using organically collected Tulu social media text containing code-mixed and mixed-script variation before downstream supervised classification. Liu et al. (2022) showed that domain adaptation can improve the performance of transformer-based language models in downstream tasks. We incorporated additional organic Tulu social media text for domain adaptation. This corpus consisted of naturally occurring Tulu social media comments and was intended to better reflect real-world code-mixed and mixed-script usage.

Unlike the labeled shared-task comments, this organic data consisted of naturally occurring Tulu social media comments collected to better reflect the linguistic variation found in real-world usage, including mixed-script and code-mixed forms. Following the data development approach described in Wong and Durward (2024), the organic Tulu data was retrieved using a Tulu language identification pipeline trained with Wikipedia-based data, which was used to identify candidate Tulu social media text for continued language-model adaptation. We used this organically collected Tulu social media text from the *Global Corpus of Language Use* (Dunn, 2020), which contains code-mixed and mixed-script variation, as unlabeled in-domain data for further adapting XLM-RoBERTa before downstream classification.

We adapted XLM-RoBERTa using a masked language modeling objective and then trained an `AutoModelForMaskedLM` model on this corpus. We trained the masked language model for two epochs with a per-device batch size of 16 and a learning rate of 5×10^{-5}. We used the AdamW optimization setup provided in the training framework (Loshchilov and Hutter, 2019). After training, we saved the adapted checkpoint and tokenizer for use in downstream supervised fine-tuning.

3.3 Supervised Classification

As discussed in Section 3.2, we developed our classification systems using both the original XLM-RoBERTa checkpoint and the organically adapted checkpoint produced during the domain adaptation stage. We then fine-tuned these models on the labeled shared-task data for the two Tulu classification settings: Task 1 (coarse-grained hope tone classification); and, Task 2 (fine-grained hope type classification). All model training was conducted in a shared Google Colab environment. Note that the relevant code notebook and scripts are available on GitHub[1]. For each task, we trained a sequence classification model using the appropriate number of output labels. We used `AutoModelForSequenceClassification` with the XLM-RoBERTa tokenizer and then trained the classifier for up to four epochs with a learning rate of 2×10^{-5}, using the same fine-tuning configuration for both the baseline and organically adapted systems. Four candidate systems overall were produced: a baseline and an organically adapted model for both Task 1 and Task 2 each. The performance of these candidate systems on the development set is reported in Section 4. Based on the development set macro average F_1-scores, we selected the organically adapted models as our submitted systems for both Task 1 and Task 2.

4 Results

The macro average F_1-scores of our candidate models on the development set are presented in Table 3 and Table 4. Both organic models performed better than the baseline models on the development set. In Task 1, the best performing candidate model was the organic model which yielded a macro average F_1-score of 0.5238 compared to 0.5227 for the baseline model. In Task 2, the best performing candidate model was also the organic model which yielded a macro average F_1-score of 0.3416 compared to 0.3171 for the baseline model. Based

[1] https://github.com/landrewi/cantnlp-HopeSpeechDetection-DravidianLangTech2026

Table 3: Validation results for Task 1: Coarse-Grained Hope Speech Classification. The best performing model based on macro average F_1-score is in **bold**.

Metric	Baseline	Organic
Accuracy	0.6269	0.6869
Macro F_1	0.5227	**0.5238**
Weighted F_1	0.6364	0.6545

Table 4: Validation results for Task 2: Fine-Grained Hope Speech Classification. The best performing model based on macro average F_1-score is in **bold**.

Metric	Baseline	Organic
Accuracy	0.4135	0.5337
Macro F_1	0.3171	**0.3416**
Weighted F_1	0.3753	0.4732

on these model performance metrics, we therefore selected the organically adapted models as our submitted systems for both Task 1 and Task 2.

5 Discussion

Based on the evaluation metrics alone, our organically adapted systems performed modestly better than the baseline systems on the development set for both tasks. With reference to Table 3 and Table 4, we see that the organically adapted model improved model performance over the baseline model in both the coarse-grained and fine-grained conditions. In Task 1, the macro average F_1-score increased from 0.5227 to 0.5238, while in Task 2 the macro average F_1-score increased from 0.3171 to 0.3416, suggesting that further adapting XLM-RoBERTa on organically collected Tulu social media text containing code-mixed and mixed-script variation provided a measurable benefit for hope speech classification, particularly for the fine-grained task (Task 2).

While the development set results indicate consistent improvement, they were not uniform across metrics and tasks. In Task 1, the macro average F_1 improvement was small, but accuracy and weighted F_1 increased more clearly, possibly reflecting that the organically adapted model better captured the majority-class patterns while still struggling with minority-class distinctions. As for Task 2, the organically adapted model improved across all reported metrics, suggesting that exposing the model to organically collected in-domain text containing code-mixed and mixed-script variation may be specifically helpful when the label space was larger and the distinctions between classes were more subtle.

Table 5: Macro average F_1-score from official test evaluation and rank for both tasks.

Task	F_1-score	Rank
Coarse-Grained	0.50	5
Fine-Grained	0.33	7

When comparing our submitted systems with the official test evaluation results found in Table 5, our organically adapted systems achieved a macro average F_1-score of 0.50 for Task 1 and 0.33 for Task 2, ranking fifth and seventh overall, respectively. The best performing team in Task 1 was Team `CUET_Synthetica` which yielded a macro average F_1-score of 0.58, while we yielded a macro average F_1-score of 0.50 and ranked 5th overall. The best performing team in Task 2 was also Team `CUET_Synthetica` which yielded a macro average F_1-score of 0.42, while we yielded a macro average F_1-score of 0.33 and ranked 7th overall. Despite this mid-ranged performance, these results indicate that the approach of adapting multilingual pretrained models using organically collected unlabeled Tulu social media text for further language-model adaptation is viable in this low-resource code-mixed setting, but the complexity of code-mixed Tulu requires more robust data or architectural innovations.

6 Conclusion

While our submitted systems did not achieve the best results, our organically adapted approach produced improvements over the baseline models on the development set for both Task 1 and Task 2. These results suggest that further adapting XLM-RoBERTa on organically collected Tulu social media text containing code-mixed and mixed-script variation can improve hope speech detection in code-mixed Tulu. At the same time, the official test results indicate that this task remains challenging, especially under class imbalance and fine-grained label distinctions.

Limitations

One limitation of the current study is that we only evaluated one domain adaptation strategy based on

organically collected Tulu social media text and did not compare it directly with a synthetic script-switching variant such as that explored in (Wong and Durward, 2024). This prior work explored both synthetic and organic script-switching for domain adaptation, whereas our current system focused only on further adapting XLM-RoBERTa using organically collected Tulu social media text containing code-mixed and mixed-script variation. As a result, we cannot determine whether the gains observed on the development set were specific to the organic adaptation pipeline or whether a synthetic approach would have produced similar improvements.

Another limitation was class imbalance in both tasks, as shown in Table 1 and Table 2. In Task 1, the training set was dominated by the *uninvolved* and *encouraging* classes, while *discouraging* and *blended tone* were underrepresented. In Task 2, *inspiring hope* and *hopelessness* appeared most frequently, while *fading hope*, *optimistic hope*, and *realistic hope* were underrepresented. We did not apply any explicit class-imbalance mitigation techniques such as class-weighted loss, focal loss, or over/under-sampling during training. Future work could test whether these strategies improve performance on minority classes, particularly for Task 2. This imbalance may reduce the model's ability to learn reliable distinctions for less frequent hope-related categories and may have contributed to the modest official test performance.

Finally, beyond the upstream and downstream impacts of bias in multilingual PLMs, we also recognize that incorporating externally collected digital language data for language-model adaptation may introduce additional biases not fully addressed in this paper, such as sampling bias, geographic bias, and variation tied to platform-specific language-use (Goldhahn et al., 2012; Dunn, 2020). These limitations suggest that future work should compare multiple adaptation strategies more directly and further examine the effects of data balance and external-data bias in code-mixed Tulu hope speech detection.

Ethical Considerations

Our system contributes to much needed research on under-resourced languages such as Tulu (Shetty, 2024). The results suggest that existing PLMs, such as XLM-RoBERTa (Conneau et al., 2020), are ill-equipped to meet the needs of Tulu language speakers. Therefore, the findings from our paper provides offers an insightful benchmark for Tulu within the area of hope speech detection. However, we need to think beyond development, but also applied to benefit Tulu speaking communities. Much like hate speech detection (Wong, 2024), we should also adopt a similar view to determine the social impact of hope speech detection research. Moreover, we should consider how systems developed can be deployed to detect hope speech in real-world social media contexts.

References

Muhammad Ahmad, Iqra Ameer, Wareesa Sharif, Sardar Usman, Muhammad Muzamil, Ameer Hamza, Muhammad Jalal, Ildar Batyrshin, and Grigori Sidorov. 2025. Multilingual hope speech detection from tweets using transfer learning models. *Scientific Reports*, 15(1):9005.

Sabur Butt, Fazlourrahman Balouchzahi, Maaz Amjad, Salud María Jiménez-Zafra, Hector G. Ceballos, and Grigori Sidorov. 2025. Overview of PolyHope at IberLEF 2025: Optimism, Expectation or Sarcasm? *Procesamiento del Lenguaje Natural*, 75(0):461–474.

Bharathi Raja Chakravarthi. 2022. Hope speech detection in YouTube comments. *Social Network Analysis and Mining*, 12(1):75.

Bharathi Raja Chakravarthi and Vigneshwaran Muralidaran. 2021. Findings of the Shared Task on Hope Speech Detection for Equality, Diversity, and Inclusion. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 61–72, Kyiv, Ukraine. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, Subalalitha Chinnaudayar Navaneethakrishnan, John Philip McCrae, Miguel Ángel García-Cumbreras, Salud María Jiménez-Zafra, Rafael Valencia-García, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Daniel García-Baena, and José Antonio García-Díaz. 2022a. Overview of the Shared Task on Hope Speech Detection for Equality, Diversity, and Inclusion. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 378–388, Dublin, Ireland. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Thenmozhi Durairaj, John McCrae, Paul Buitelaar, Prasanna Kumaresan, and Rahul Ponnusamy. 2022b. Overview of The Shared Task on Homophobia and Transphobia Detection in Social Media Comments. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 369–377, Dublin, Ireland. Association for Computational Linguistics.

Anusha Chhabra and Dinesh Kumar Vishwakarma. 2023. A literature survey on multimodal and multilingual automatic hate speech identification. *Multimedia Systems*, 29(3):1203–1230.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. 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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Jonathan Dunn. 2020. Mapping languages: the Corpus of Global Language Use. *Language Resources and Evaluation*, 54(4):999–1018. Https://doi.org/10.1007/s10579-020-09489-2.

Daniel García-Baena, Fazlourrahman Balouchzahi, Sabur Butt, Miguel Ángel García-Cumbreras, Atnafu Lambebo Tonja, José Antonio García-Díaz, Selen Bozkurt, Bharathi Raja Chakravarthi, Hector G. Ceballos, Rafael Valencia-García, Grigori Sidorov, L. Alfonso Ureña-López, Alexander Gelbukh, and Salud María Jiménez-Zafra. 2024. Overview of HOPE at IberLEF 2024: Approaching Hope Speech Detection in Social Media from Two Perspectives, for Equality, Diversity and Inclusion and as Expectations. *Procesamiento del Lenguaje Natural*, 73(0):407–419.

Sayani Ghosal, Devendra Kumar Tayal, and Amita Jain. 2026. Hope WVBL: Improving Automatic Hope Speech Detection Using Word Embedding and Bidirectional LSTM. In *Proceedings of International Conference on Information Technology and Intelligence*, pages 131–142, Singapore. Springer Nature.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Eftekhar Hossain, Omar Sharif, and Mohammed Moshiul Hoque. 2021. NLP-CUET@LT-EDI-EACL2021: Multilingual Code-Mixed Hope Speech Detection using Cross-lingual Representation Learner. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 168–174, Kyiv. Association for Computational Linguistics.

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text Classification Algorithms: A Survey. *Information*, 10(4):150.

Prasanna Kumar Kumaresan, Bharathi Raja Chakravarthi, Subalalitha Cn, Miguel Ángel García-Cumbreras, Salud María Jiménez Zafra, José Antonio García-Díaz, Rafael Valencia-García, Momchil Hardalov, Ivan Koychev, Preslav Nakov, Daniel García-Baena, and Kishore Kumar Ponnusamy. 2023. Overview of the Shared Task on Hope Speech Detection for Equality, Diversity, and Inclusion. In *Proceedings of the Third Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 47–53, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Kirti Kumari, Shirish Shekhar Jha, Zarikunte Kunal Dayanand, and Praneesh Sharma. 2023. ML&AI_iiitranchi@LT-EDI-2023: Identification of Hope Speech of YouTube comments in Mixed Languages. In *Proceedings of the Third Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 214–222, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Xianbo Li. 2022. Text Classification of Conversational Implicatures Based on Lexical Features. *Applied Artificial Intelligence*, 36(1):2127598.

Xiaofeng Liu, Chaehwa Yoo, Fangxu Xing, Hyejin Oh, Georges El Fakhri, Je-Won Kang, and Jonghye Woo. 2022. Deep Unsupervised Domain Adaptation: A Review of Recent Advances and Perspectives. *APSIPA Transactions on Signal and Information Processing*, 11(1).

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *Proceedings in the Seventh International Conference on Learning Representations*, New Orleans, LA.

Deepawali Sharma, Vedika Gupta, Vivek Kumar Singh, and Bharathi Raja Chakravarthi. 2025. Stop the Hate, Spread the Hope: An Ensemble Model for Hope Speech Detection in English and Dravidian Languages. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*

Megha Sharma and Gaurav Arora. 2021. Spartans@LT-EDI-EACL2021: Inclusive Speech Detection using Pretrained Language Models. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 188–192, Kyiv. Association for Computational Linguistics.

Poorvi Shetty. 2024. Natural Language Processing for Tulu: Challenges, Review and Future Scope. In *Speech and Language Technologies for Low-Resource Languages*, pages 93–109, Cham. Springer Nature Switzerland.

Priyadharshini Thandavamurthi, Samyuktaa Sivakumar, Shwetha Sureshnathan, Thenmozhi D., Bharathi B,

and Gayathri Gl. 2023. TERCET@LT-EDI-2023: Hope Speech Detection for Equality, Diversity, and Inclusion. In *Proceedings of the Third Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 257–261, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Durairaj Thenmozhi, Rathnakar Shetty P, Parameshwar R. Hegde, Anusha M D, Raksha Adyanthaya, Mohammed Fadhel Aljunid, Prasanna Kumar Kumaresan, and Bharathi Raja Chakravarthi. 2026. Findings of the Shared Task on Hope Speech Detection in Tulu. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, San Diego, CA. Association for Computational Linguistics.

Sidney Wong and Matthew Durward. 2024. cantnlp@LT-EDI-2024: Automatic Detection of Anti-LGBTQ+ Hate Speech in Under-resourced Languages. In *Proceedings of the Fourth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 177–183, St. Julian's, Malta. Association for Computational Linguistics.

Sidney Gig-Jan Wong. 2024. What is the social benefit of hate speech detection research? A Systematic Review. In *Proceedings of the Third Workshop on NLP for Positive Impact*, pages 1–12, Miami, Florida, USA. Association for Computational Linguistics.

CHMOD_777@DravidianLangTech 2026: Context-Aware Fine-tuned MuRIL for Abusive Tamil Text Detection on Social Media

Arunaggiri Pandian Karunanidhi[1], **Prabalakshmi Arumugam**[2]
[1]Micron Technology, United States
[2]Boise State University, United States
arunaggiri.karunanidhi@gmail.com, prabalakshmiarumugam410@gmail.com

Abstract

This paper describes Team CHMOD_777's system for the DravidianLangTech@ACL 2026 shared task on detecting abusive Tamil text targeting women on social media. We fine-tune three transformer backbones (MuRIL (Multilingual Representations for Indian Languages), XLM-RoBERTa, IndicBERT-v3) with Focal Loss and weighted sampling, systematically evaluating the effects of context length, hyperparameter tuning, and language-specific pre-training. Our best system, MuRIL with 256-token context, achieves 82.76% Macro F1 on the development set and 80.61% on the official test set, ranking 6th out of 24 teams. We find that (1) extending context from 128 to 256 tokens improves F1 while converging 2.4× faster, (2) language-specific pre-training (MuRIL, 236M) outperforms larger models (IndicBERT, 270M), and (3) default hyperparameters are optimal, with every tuning attempt degrading performance.

1 Introduction

Online abuse targeting women on social media is a pervasive problem across languages and cultures. In low-resource languages such as Tamil, the challenge is amplified by code-mixing, transliteration, and culturally specific expressions of abuse. The DravidianLangTech@ACL 2026 shared task (Rajiakodi et al., 2026) addresses this by framing the detection of abusive Tamil text targeting women as a binary classification problem.

The task presents three challenges: (1) code-mixed Tamil-English text requiring multilingual tokenization; (2) transliteration where Tamil is written in Latin script on social media; and (3) culturally specific patterns of abuse with euphemisms that may not transfer from other languages.

Our approach is deliberately simple: fine-tuning pre-trained transformer models with Focal Loss (Lin et al., 2017) and weighted sampling, without custom architectural modifications. We focus instead on systematic evaluation of context length (128 vs. 256 tokens), model selection (language-specific vs. multilingual), and hyperparameter sensitivity. Our contributions are:

- MuRIL (Multilingual Representations for Indian Languages) with 256-token context achieves 82.76% Macro F1, ranking 6th out of 24 teams;
- Extending context from 128 to 256 tokens improves F1 while converging 2.4× faster;
- Language-specific pre-training outperforms larger general models, and default hyperparameters prove optimal.[1]

2 Related Work

Detecting abusive and hateful content on social media has received significant attention in NLP. Early approaches relied on handcrafted features and traditional machine learning classifiers (Waseem and Hovy, 2016), while recent work has shifted to transformer-based models that capture contextual nuances. For Indian languages specifically, shared tasks at DravidianLangTech workshops (Chakravarthi et al., 2022; Bharathi et al., 2022) have driven progress in code-mixed abuse detection and speech processing for Tamil, Telugu, and Malayalam, with transformer-based systems consistently outperforming feature-engineered baselines.

Tamil social media text poses unique challenges due to extensive code-mixing with English and frequent transliteration from Tamil script to Latin characters. Jada et al. (2021) demonstrated that transformer architectures significantly outperform traditional methods for Dravidian language sentiment analysis. Sai Kumar et al. (2021) showed that

[1]The code for this work is available at: https://github.com/Arunaggiri-Pandian/Abusive_Tamil_Text_Classification

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 176–180
July 3, 2026 ©2026 Association for Computational Linguistics

Split	Non-Abusive	Abusive	Total	Abusive %
Train	1,694	1,592	3,286	48.4
Dev	189	177	366	48.4
Test	472	441	913	48.3

Table 1: Dataset statistics. Nearly balanced (1:1 ratio).

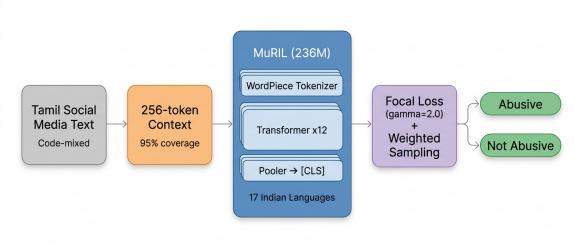


Figure 1: System pipeline: input text tokenized to 256 tokens, encoded by MuRIL, classified with Focal Loss.

BERT-based models achieve competitive performance for tweet-level classification tasks involving code-mixed text. Prior work on multimodal Dravidian content analysis (Premjith et al., 2022) has further established that pre-trained language models can capture cross-lingual patterns in mixed-script settings.

Among pre-trained models, MuRIL (Khanuja et al., 2021), pre-trained on 17 Indian languages with transliteration augmentation, has shown strong results for Tamil. XLM-RoBERTa (Conneau et al., 2020) provides broad multilingual coverage but dilutes language-specific representations across 100+ languages. IndicBERT-v3 (Doddapaneni et al., 2023) targets Indian languages specifically but with a different pre-training strategy. Our work provides a direct comparison of these three paradigms on the same task and dataset.

Focal Loss (Lin et al., 2017), originally proposed for class-imbalanced object detection, down-weights well-classified examples through a focusing parameter γ and concentrates learning on hard examples near the decision boundary. While primarily used for imbalanced datasets, we demonstrate its utility even for nearly balanced binary classification where ambiguous cases benefit from focused training.

3 Methodology

3.1 Dataset

The dataset comprises 3,286 training, 366 development, and 913 test samples. Table 1 shows the class distribution. The dataset is nearly balanced (1:1 ratio) with identical proportions across train and dev splits, indicating careful stratified splitting.

Tamil social media text presents specific challenges. Code-mixing is highly prevalent, with tweets frequently alternating between Tamil and English within a single utterance. Transliteration adds further complexity as Tamil is often written using Latin characters (e.g., "romba bayama iruku" instead of native Tamil script). The average text length is approximately 111 characters (14 words), though abusive content tends to use longer expressions with multiple clauses.

3.2 Model Architecture

We adopt a straightforward classification approach: each pre-trained encoder maps an input text to a pooled `[CLS]` representation $\mathbf{h} \in R^d$, which is classified via a dropout layer (p=0.1) and a linear head $W \in R^{d \times 2}$ (Devlin et al., 2019). We compare three backbones that represent different pre-training strategies, all sharing the same hidden dimension (d=768) and 12-layer architecture:

- **MuRIL** (Khanuja et al., 2021): 236M params, pre-trained on 17 Indian languages with transliteration augmentation, providing native Tamil coverage including romanized text.
- **XLM-RoBERTa** (Conneau et al., 2020): 278M params, pre-trained on 100+ languages via CommonCrawl, offering broad but diluted per-language coverage.
- **IndicBERT-v3** (Doddapaneni et al., 2023): 270M params, pre-trained on 11 Indic languages with IndicCorp data, using a different pre-training strategy than MuRIL.

All models are fine-tuned end-to-end using HuggingFace Transformers. A key design decision is the input context length: we evaluate 128 and 256 tokens to study how context affects Tamil abuse detection. We chose 128 as the standard BERT default and 256 as a practical upper bound that covers 95% of samples in the dataset without excessive padding. Longer contexts (384, 512) would introduce majority-padding tokens that dilute attention, particularly problematic for the shorter social media texts in this corpus. Figure 1 shows the overall pipeline.

3.3 Training Configuration

We use Focal Loss (Lin et al., 2017) with γ=2.0 and inverse-frequency class weights α_t:

$$\text{FL}(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (1)$$

Parameter	Value
Max sequence length	256 (or 128 for ablation)
Batch size	16
Learning rate	1e-5
Optimizer	AdamW
Weight decay	0.01
Loss function	Focal (γ=2.0)
Gradient clipping	max_norm = 1.0
Early stopping patience	10 epochs
Max epochs	50
Seed	42

Table 2: Training hyperparameters (MuRIL, 256 tokens).

Model	Tokens	MF1	Epoch	Δ
MuRIL v2	**256**	**82.76**	**9**	
MuRIL v1	128	82.50	22	−0.26
MuRIL Tuned	128	82.45	22	−0.31
MuRIL Tuned+GA	128	82.23	17	−0.53
XLM-RoBERTa	256	81.95	10	−0.81
IndicBERT-v3	256	74.02	39	−8.74

Table 3: Dev set results. Tuned = lr 2e-5, warmup, γ=1.0. GA = gradient accumulation.

where p_t is the predicted probability for the true class. The focusing term $(1 - p_t)^\gamma$ down-weights well-classified examples, concentrating learning on hard examples near the decision boundary. Even though the dataset is nearly balanced, this is valuable for abuse detection where the distinction between abusive and non-abusive content is often subjective. WeightedRandomSampler ensures balanced mini-batches, and early stopping monitors dev Macro F1 with patience of 10 epochs. Table 2 lists the full configuration.

To assess hyperparameter sensitivity, we train additional MuRIL variants with increased learning rate (2e-5), linear warmup (10%), higher weight decay (0.02), reduced focal gamma (1.0), and gradient accumulation (effective batch size 32). These experiments allow us to isolate the impact of each training decision.

4 Results

4.1 Development Set Performance

Table 3 presents all six experiments. MuRIL with 256-token context achieves the best Macro F1 of 82.76%, converging at epoch 9.

4.2 Per-Class Analysis

Table 4 shows per-class metrics for the best model. The F1 gap between classes is only 1.41 percentage points, demonstrating symmetric performance.

Class	Precision	Recall	F1
Non-Abusive	82.81	84.13	83.46
Abusive	82.76	81.36	82.05
Macro Avg	**82.79**	**82.74**	**82.76**

Table 4: Per-class metrics (%) for MuRIL v2 (256 tokens).

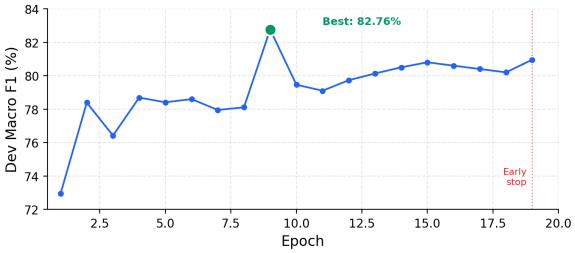


Figure 2: Training progression. F1 peaks at epoch 9.

The largest asymmetry is in recall (2.77 point gap), where the model is slightly better at identifying Non-Abusive text, likely reflecting the inherent difficulty of detecting subtler forms of abuse.

4.3 Training Dynamics

Figure 2 shows the training progression for MuRIL v2. The model exhibits rapid initial learning (F1 jumps from 72.95% to 78.39% in one epoch), peaks at epoch 9 (82.76%), and remains stable at 80.95% by epoch 19 when early stopping triggers. MuRIL v2 retains within 1.81 points of its peak over 10 additional epochs, indicating effective regularization from Focal Loss and the balanced sampler.

4.4 Official Test Set Results

On the official test set, our primary submission achieves 80.61% Macro F1, ranking 6th out of 24 teams. Table 5 compares dev and test per-class metrics. The symmetric performance holds on the test set, with only 0.68% F1 gap between classes (Non-Abusive 80.95%, Abusive 80.27%). The competition was highly competitive, with the top 10 spanning only 2.96 points. Full leaderboard results are available in the task overview paper (Rajiakodi et al., 2026).

5 Analysis

Extending context from 128 to 256 tokens yields the most impactful improvement: MuRIL (256 tokens, F1=82.76%) outperforms its 128-token variant (F1=82.50%) while converging 2.4× faster (9 vs 22 epochs), as shown in Figure 3. The extended

Class	Dev F1	Test P	Test R	Test F1
Non-Abusive	83.46	82.28	79.66	80.95
Abusive	82.05	78.95	81.63	80.27
Macro	**82.76**	**80.61**	**80.65**	**80.61**

Table 5: Per-class dev vs test (%). P = Precision, R = Recall.

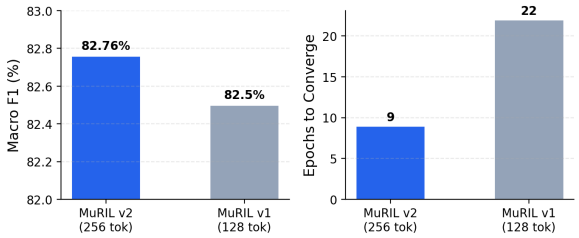


Figure 3: Context length ablation: 256 tokens achieve higher F1 with 2.4× faster convergence.

context captures complete code-mixed expressions that frequently exceed 128 tokens.

MuRIL (236M, F1=82.76%) outperforms XLM-RoBERTa (278M, 81.95%) and IndicBERT-v3 (270M, 74.02%) despite being smallest, confirming that pre-training strategy matters more than scale. Every hyperparameter tuning attempt degraded performance (largest: gradient accumulation −0.53%), as the small dataset (3,286 samples) means pre-trained weights already provide a well-conditioned loss landscape where conservative defaults are optimal. Figure 4 compares all backbones.

Error analysis reveals two failure modes: (1) subtle abuse conveyed through sarcasm or cultural euphemisms without explicit keywords, and (2) code-mixed texts where the abusive signal spans Tamil and English fragments. Focal Loss produces symmetric per-class performance (1.41% F1 gap) despite near-balanced data, confirming its value for focusing on ambiguous boundary cases.

6 Conclusion

We presented a system for detecting abusive Tamil text achieving 80.61% Macro F1 (Rank 6/24). Key findings: (1) 256-token context improves F1 and converges 2.4× faster; (2) MuRIL outperforms larger models by up to 8.74 points; (3) default hyperparameters are optimal on small datasets; (4) Focal Loss benefits even balanced classification.

Limitations

Data augmentation techniques tailored to Tamil text could yield further improvements. Ensemble

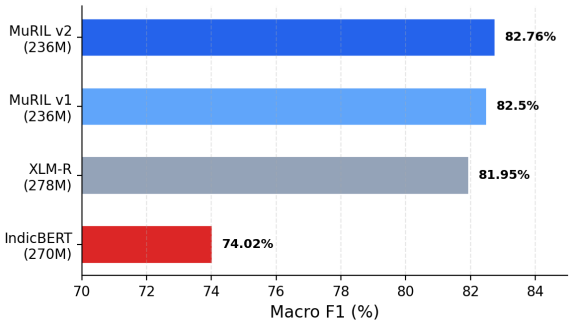


Figure 4: Model comparison. MuRIL achieves the best F1 despite fewest parameters.

methods combining MuRIL with XLM-RoBERTa may leverage their complementary strengths across different text types. Incorporating external lexical resources for Tamil abusive language is another avenue that could improve recall on subtle forms of abuse.

References

B. Bharathi, Bharathi Raja Chakravarthi, Subalalitha Cn, N. Sripriya, Arunaggiri Pandian Karunanidhi, and Swetha Valli. 2022. Findings of the shared task on speech recognition for vulnerable individuals in Tamil. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Prasanna Kumar Kumaresan, Rahul Ponnusamy, John P. McCrae, Elizabeth Sherly, and Saranya Rajiakodi. 2022. Findings of the shared task on sentiment analysis in Tamil and Telugu code-mixed text. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*. 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. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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 (NAACL-HLT)*. Association for Computational Linguistics.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M Khapra, Divyanshu Kakwani, Anoop Kumar, Jessica Murber, Preethi

Nemani, and Ravi Shankar Lakumarapu. 2023. Towards leaving no Indic language behind: Building monolingual corpora, benchmark and models for Indic languages. *arXiv preprint arXiv:2212.05409*.

Pawan Kalyan Jada, D. Sashidhar Reddy, Konthala Yasaswini, Arunaggiri Pandian Karunanidhi, Prabakaran Chandran, Anbukkarasi Sampath, and Sathiyaraj Thangasamy. 2021. Transformer based sentiment analysis in Dravidian languages. In *Proceedings of FIRE 2021 (Working Notes)*.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagisetty, Shachi Gupta, Ankur Parikh, and Partha Talukdar. 2021. MuRIL: Multilingual representations for Indian languages. *arXiv preprint arXiv:2103.10730*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

B. Premjith, Bharathi Raja Chakravarthi, Malliga Subramanian, B. Bharathi, K.P. Soman, V. Dhanalakshmi, K. Sreelakshmi, Arunaggiri Pandian Karunanidhi, and Prasanna Kumaresan. 2022. Findings of the shared task on multimodal sentiment analysis and troll meme classification in Dravidian languages. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Saranya Rajiakodi, Shunmuga Priya Muthusamy Chinnan, Rajalakshmi R., Kathiravan Pannerselvam, Bhuvaneswari Sivagnanam, Jananayagan V, Charmathi Rajkumar, R Ramesh Kannan, and Bharathi Raja Chakravarthi. 2026. From Comments to Harm: A Findings Report on Abusive Tamil Text Targeting Women on Social Media Shared Task - DravidianLangTech@ACL 2026. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

T.S. Sai Kumar, K. Arunaggiri Pandian Karunanidhi, S. Thabasum Aara, and K. Nagendra Pandian. 2021. A reliable technique for sentiment analysis on tweets via machine learning and BERT. In *2021 Asian Conference on Innovation in Technology (ASIANCON)*. IEEE.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*. Association for Computational Linguistics.

CHMOD_777@DravidianLangTech 2026: LLM Augmented Transformer Fine-tuning for Tamil Political Sentiment Analysis

Arunaggiri Pandian Karunanidhi[1], **Prabalakshmi Arumugam**[2]
[1]Micron Technology, United States
[2]Boise State University, United States
arunaggiri.karunanidhi@gmail.com, prabalakshmiarumugam410@gmail.com

Abstract

This paper describes Team CHMOD_777's system for the DravidianLangTech@ACL 2026 shared task on political multiclass sentiment analysis of Tamil Twitter comments. The task requires classifying Tamil political tweets into seven sentiment categories under severe class imbalance (8:1 ratio). We address this challenge through LLM-based data augmentation using Gemini 2.5 Flash, expanding training data from 4,352 to 15,316 samples (3.5× the original). Our best system, MuRIL fine-tuned on augmented data with Focal Loss (γ=3.0) and weighted sampling, achieves 35.79% Macro F1 on the development set, a 67% relative improvement over the non-augmented baseline. On the official test set, our system achieves 34.25% Macro F1, ranking 12th out of 22 participating teams. We find that (1) language-specific pre-training (MuRIL, 236M) outperforms larger general models (IndicBERT-v3, 1B), (2) smaller models benefit disproportionately from augmentation, and (3) Substantiated is the hardest category (F1=10.7%) due to its requirement for factual reasoning.

1 Introduction

Political sentiment analysis on social media presents unique challenges: highly imbalanced class distributions, code-mixed text, and culturally-specific sentiment expressions. The DravidianLangTech@ACL 2026 shared task (Chakravarthi et al., 2026) requires classifying Tamil political tweets into seven fine-grained categories: Opinionated (31.3%), Sarcastic (18.2%), Neutral (14.6%), Positive (13.2%), Substantiated (9.5%), Negative (9.3%), and None of the above (3.9%).

Three key challenges distinguish this task: (1) severe class imbalance with an 8:1 ratio between the majority (Opinionated, 1,361 samples) and minority (None of the above, 171 samples) classes; (2) code-mixing where 84.9% of tweets combine Tamil and English; and (3) fine-grained distinctions between semantically similar categories such as Opinionated vs. Sarcastic.

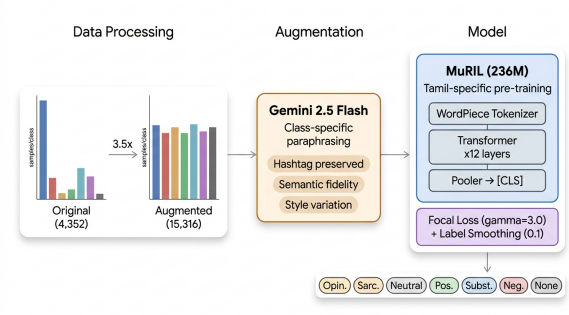


Figure 1: LLM-augmented pipeline. Original data (4,352 samples) is expanded 3.5x via Gemini 2.5 Flash, then used to fine-tune MuRIL with Focal Loss (γ=3.0) for 7-class classification.

Our approach centers on LLM-based data augmentation to address data scarcity for minority classes, combined with Focal Loss (Lin et al., 2017) and weighted sampling during training. We demonstrate that augmentation with Gemini 2.5 Flash yields a 67% relative improvement in Macro F1, and that language-specific pre-training (MuRIL) outperforms larger multilingual models. Figure 1 illustrates the three-stage pipeline.[1]

2 Related Work

Transformer-based models have achieved strong results on Dravidian language tasks (Jada et al., 2021; Sai Kumar et al., 2021). MuRIL (Khanuja et al., 2021), pre-trained on 17 Indian languages with transliteration augmentation, has shown particular effectiveness for Tamil. Previous DravidianLangTech shared tasks (Chakravarthi et al., 2022; Premjith et al., 2022) and the prior edition of this task (Chakravarthi et al., 2025) have explored sen-

[1]The code for this work is available at: https://github.com/Arunaggiri-Pandian/Political_Multiclass_Sentiment_Analysis_in_Tamil

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 181–185
July 3, 2026 ©2026 Association for Computational Linguistics

Class	Train	%	Augmented
Opinionated	1,361	31.3	2,717
Sarcastic	790	18.2	2,366
Neutral	637	14.6	2,487
Positive	575	13.2	2,256
Substantiated	412	9.5	2,392
Negative	406	9.3	2,426
None of above	171	3.9	672
Total	**4,352**		**15,316**

Table 1: Class distribution before and after LLM-based augmentation (3.5× expansion).

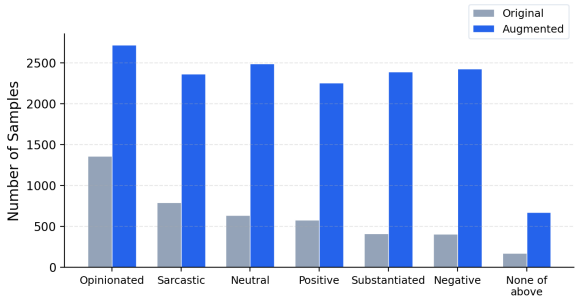


Figure 2: Impact of LLM-based augmentation on class distribution. Augmentation reduces the imbalance ratio from 8:1 to 4:1.

timent analysis and content classification for code-mixed Dravidian text and established baselines for Tamil political sentiment classification.

For class imbalance, Focal Loss (Lin et al., 2017) down-weights well-classified examples, focusing training on harder cases. Data augmentation via LLM paraphrasing (Dai and Adel, 2020; Kumar et al., 2020) has emerged as an effective strategy, particularly for low-resource languages where rule-based techniques like EDA (Wei and Zou, 2019) fail to preserve grammatical structure.

3 Methodology

3.1 Dataset

The dataset comprises 4,352 training, 544 development, and 544 test samples. Table 1 shows the severe class imbalance across the seven categories. Analysis reveals that 84.9% of tweets are code-mixed (Tamil+English), 90.8% contain political hashtags, and the average length is 172 characters. Figure 2 illustrates the class distribution and augmentation impact.

3.2 LLM-Based Data Augmentation

Traditional augmentation techniques (random deletion, swap, insertion) proved ineffective for Tamil, breaking grammatical structure and altering meaning. We instead implemented LLM-based paraphrasing using Gemini 2.5 Flash via Google Cloud Vertex AI with temperature 0.9.

Our pipeline applies class-specific variation counts targeting ∼2,000 samples per class: 1× for Opinionated, 2× for Sarcastic/Neutral, 3× for Positive, and 4× for Substantiated/Negative. We skip augmentation for None of the above as 45.6% of its samples contain fewer than 5 words, making meaningful paraphrasing infeasible.

Quality control uses a three-layer hashtag preservation strategy: (1) explicit prompt instructions to copy hashtags verbatim, (2) dynamic injection of the exact hashtag list per tweet into the prompt, and (3) post-generation validation rejecting samples with missing hashtags. Additional checks ensure output length falls within 50% to 200% of the original and is not identical to the source. This reduces the imbalance ratio from 8:1 to 4:1.

3.3 Model Architecture

Our classification pipeline follows the standard fine-tuning paradigm (Devlin et al., 2019). Given an input tweet x, the pre-trained encoder produces a sequence of contextualized token representations. We extract the pooled representation $\mathbf{h} \in R^d$ from the `[CLS]` token. This representation is passed through a dropout layer (p=0.1) followed by a linear classification head $W \in R^{d \times 7}$ to produce the output logits:

$$\mathbf{y} = W \cdot \text{Dropout}(\mathbf{h}) + \mathbf{b} \quad (1)$$

where d is the hidden dimension of the encoder (768 for MuRIL, 1024 for IndicBERT-1B). We evaluate three pre-trained transformer backbones:

- **MuRIL** (Khanuja et al., 2021): 236M params, BERT-base architecture (d=768, 12 layers, 12 heads), pre-trained on 17 Indian languages including Tamil with transliteration augmentation for handling romanized text.
- **IndicBERT-v3-1B** (Doddapaneni et al., 2023): 1B params, BERT-large architecture (d=1024, 24 layers, 16 heads), pre-trained on 11 Indic languages with larger corpora.
- **IndicBERT-v3-270M**: 270M params, medium-sized variant (d=768, 12 layers), same pre-training data as the 1B model.

All models are loaded via HuggingFace Transformers and fine-tuned end-to-end, updating all

Parameter	Value
Max sequence length	256
Batch size	16
Gradient accumulation	2 (eff. batch 32)
Learning rate	1e-5
Optimizer	AdamW
Weight decay	0.01
Warmup ratio	0.15
Loss function	Focal (γ=3.0)
Label smoothing	0.1
Early stopping patience	10 epochs
Max epochs	50
Seed	42

Table 2: Training hyperparameters for our best model (MuRIL on augmented data).

encoder parameters along with the classification head.

3.4 Training Configuration

To address class imbalance, we combine Focal Loss (Lin et al., 2017) with γ=3.0 and WeightedRandomSampler, which together ensure the model focuses on hard minority-class examples while maintaining balanced mini-batches. We set γ=3.0 (higher than the standard 2.0) due to the severe 8:1 class imbalance, which requires more aggressive down-weighting of majority-class examples. Label smoothing (ϵ=0.1) prevents overconfident predictions on augmented data that may not perfectly represent real distributions. Both values were selected based on preliminary experiments on the dev set; we did not perform exhaustive grid search due to computational constraints. We use early stopping on dev Macro F1 with patience of 10 epochs. Table 2 summarizes the full training configuration.

4 Results

4.1 Development Set Performance

Table 3 shows performance across all model configurations. All epoch counts reflect early stopping on dev Macro F1 (patience=10), not fixed training duration. MuRIL on augmented data achieves the best Macro F1 of 35.79%.

4.2 Per-Class Analysis

Table 4 presents per-class metrics for our best model. Performance varies dramatically: None of the above achieves 86.5% F1 while Substantiated reaches only 10.7%.

Model	Data	MF1	Epoch
MuRIL (236M)	Orig.	21.44	10
IndicBERT-1B	Orig.	30.28	13
MuRIL (236M)	**Aug.**	**35.79**	**39**
IndicBERT-1B	Aug.	32.09	8
IndicBERT-270M	Aug.	27.53	22

Table 3: Dev set Macro F1 (%) and best epoch. Augmented = 15,316 samples via Gemini.

Class	Precision	Recall	F1
None of above	94.1	80.0	86.5
Sarcastic	41.5	44.3	42.9
Opinionated	33.9	36.6	35.2
Neutral	29.3	32.1	30.7
Positive	22.6	27.5	24.8
Negative	22.5	17.6	19.8
Substantiated	17.4	7.7	10.7
Macro Avg	**37.3**	**35.1**	**35.8**

Table 4: Per-class precision (P), recall (R), and F1 (%) for MuRIL trained on augmented data, evaluated on the dev set.

4.3 Official Test Set Results

On the official test set, our primary submission achieves 34.25% Macro F1, ranking 12th out of 22 teams. Table 5 compares per-class performance between dev and test sets. The overall dev-to-test gap is only 1.54 points, confirming robust generalization. None of the above remains the strongest class (95.83% test F1), while Negative and Substantiated remain the weakest. Full leaderboard results are available in the task overview paper (Chakravarthi et al., 2026).

5 Analysis

Augmentation is the most impactful factor: MuRIL improves from 21.44% to 35.79% F1 (+67%), while IndicBERT-1B gains only +6% (Figure 3). MuRIL (236M) outperforms IndicBERT-v3 (1B) by 3.7 points, confirming domain-specific pretraining outweighs scale (Jada et al., 2021). Per-class analysis (Figure 4) shows None of the above is easiest (86.5% F1, short distinctive tweets) while Substantiated is hardest (10.7% F1, requires factual reasoning).

The training F1 reaches 0.89 while dev F1 plateaus at 0.36 (gap = 0.53), indicating overfitting to synthetic patterns: LLM paraphrases share stylistic regularities that the model memorizes rather than learning generalizable boundaries. The modest dev-to-test drop (1.54 points) confirms that real

Class	Dev F1	Test P	Test R	Test F1
None of above	86.5	100.0	92.0	95.8
Sarcastic	42.9	38.7	38.7	38.7
Opinionated	35.2	36.8	37.4	37.1
Neutral	30.7	19.1	30.0	23.3
Positive	24.8	25.6	26.7	26.1
Substantiated	10.7	16.7	7.8	10.7
Negative	19.8	10.3	6.5	8.0
Macro	**35.8**	**35.3**	**34.2**	**34.2**

Table 5: Per-class comparison between dev and test sets (%). P = Precision, R = Recall. None of the above improves to 95.8%, while Negative presents the most challenge (8.0% test F1).

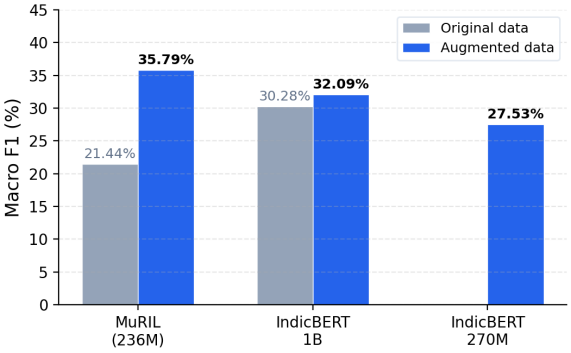


Figure 3: Augmentation impact. MuRIL benefits most (+67%).

data distributions are consistent while diverging from synthetic training. The performance ceiling (34.25%) reflects inherent difficulty of 7-class fine-grained political sentiment, particularly distinguishing Sarcastic from Opinionated and identifying Substantiated claims requiring factual reasoning.

6 Conclusion

We presented CHMOD_777's system for Tamil political sentiment classification at DravidianLangTech@ACL 2026. Our key contributions are: (1) LLM-based augmentation using Gemini 2.5 Flash with three-layer hashtag preservation improves Macro F1 by 67%; (2) language-specific pre-training (MuRIL, 236M) outperforms larger models (IndicBERT, 1B); and (3) per-class analysis reveals that Substantiated (F1=10.7%) requires factual reasoning beyond current capabilities. Our system achieves 34.25% Macro F1 on the test set (Rank 12/22), with only a 1.54-point dev-to-test drop.

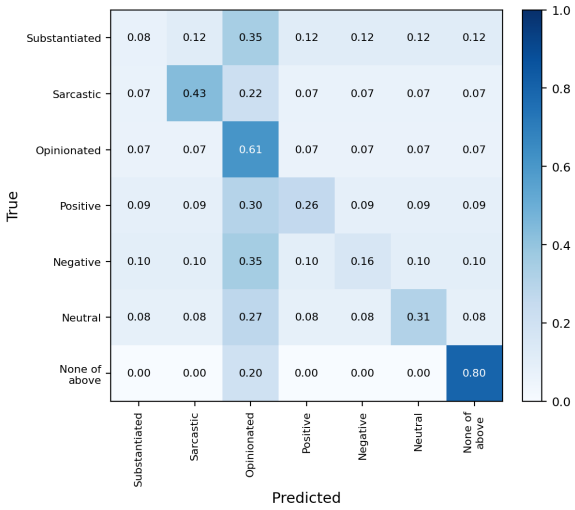


Figure 4: Confusion matrix (dev). Substantiated and Negative are most confused with Opinionated.

Limitations

Paraphrase quality was not systematically evaluated; the train-dev gap suggests augmented samples may introduce stylistic artifacts. Ensemble methods, automatic quality filtering, and external knowledge for the Substantiated class are promising future directions.

References

Bharathi Raja Chakravarthi, Prasanna Kumar Kumaresan, Rahul Ponnusamy, John P. McCrae, Elizabeth Sherly, and Saranya Rajiakodi. 2022. Findings of the shared task on sentiment analysis in Tamil and Telugu code-mixed text. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Saranya Rajiakodi, Thenmozhi Durairaj, Sathiyaraj Thangasamy, Ratnasingam Sakuntharaj, Prasanna Kumar Kumaresan, Kishore Kumar Ponnusamy, and Arunaggiri Pandian Karunanidhi. 2025. Overview 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.

Bharathi Raja Chakravarthi, Saranya Rajiakodi, Thenmozhi Durairaj, Sathiyaraj Thangasamy, Ratnasingam Sakuntharaj, Prasanna Kumar Kumaresan, Kishore Kumar Ponnusamy, and Arunaggiri Pandian Karunanidhi. 2026. Political multiclass sentiment analysis of Tamil X (Twitter) comments - DravidianLangTech@ACL 2026. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Tech-*

nologies for Dravidian Languages. Association for Computational Linguistics.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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 (NAACL-HLT)*. Association for Computational Linguistics.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M Khapra, Divyanshu Kakwani, Anoop Kumar, Jessica Murber, Preethi Nemani, and Ravi Shankar Lakumarapu. 2023. Towards leaving no Indic language behind: Building monolingual corpora, benchmark and models for Indic languages. *arXiv preprint arXiv:2212.05409*.

Pawan Kalyan Jada, D. Sashidhar Reddy, Konthala Yasaswini, Arunaggiri Pandian Karunanidhi, Prabakaran Chandran, Anbukkarasi Sampath, and Sathiyaraj Thangasamy. 2021. Transformer based sentiment analysis in Dravidian languages. In *Proceedings of FIRE 2021 (Working Notes)*.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagisetty, Shachi Gupta, Ankur Parikh, and Partha Talukdar. 2021. MuRIL: Multilingual representations for Indian languages. *arXiv preprint arXiv:2103.10730*.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. In *Proceedings of the 2020 Workshop on Living NLP*. Association for Computational Linguistics.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

B. Premjith, Bharathi Raja Chakravarthi, Malliga Subramanian, B. Bharathi, K.P. Soman, V. Dhanalakshmi, K. Sreelakshmi, Arunaggiri Pandian Karunanidhi, and Prasanna Kumaresan. 2022. Findings of the shared task on multimodal sentiment analysis and troll meme classification in Dravidian languages. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

T.S. Sai Kumar, K. Arunaggiri Pandian Karunanidhi, S. Thabasum Aara, and K. Nagendra Pandian. 2021. A reliable technique for sentiment analysis on tweets via machine learning and BERT. In *2021 Asian Conference on Innovation in Technology (ASIANCON)*. IEEE.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

CHMOD_777@DravidianLangTech 2026: Tamil-Adapted Whisper and MMS for Dialect Speech Recognition and Classification

Arunaggiri Pandian Karunanidhi[1], **Prabalakshmi Arumugam**[2]
[1]Micron Technology, United States
[2]Boise State University, United States
arunaggiri.karunanidhi@gmail.com, prabalakshmiarumugam410@gmail.com

Abstract

This paper describes Team CHMOD_777's system for the DravidianLangTech@ACL 2026 shared task on Tamil dialect speech recognition and classification. The task comprises two subtasks: classifying Tamil speech into four regional dialects (Northern, Southern, Western, Central) and transcribing dialectal Tamil speech to text. For dialect classification, we fine-tune MMS-1b-all with Focal Loss and weighted sampling, achieving 83.04% Macro F1 on the development set (5th out of 11 teams on the test set). For speech recognition, we fine-tune a Tamil-specific Whisper model (763M parameters), achieving 53.72% WER on the development set and 49.75% on the official test set, ranking 1st out of 13 teams. Our key finding is that domain-specific pre-training significantly outperforms larger general-purpose models: Tamil Whisper (763M) beats Whisper-large-v3 (1.5B) by 8 WER points despite having half the parameters.

1 Introduction

Tamil, spoken by over 80 million people, exhibits significant regional variation across four major dialect groups: Northern, Southern, Western, and Central. These dialects differ in phonology, vocabulary, and prosody, posing challenges for both dialect identification and automatic speech recognition (ASR). The DravidianLangTech@ACL 2026 shared task (Bharathi et al., 2026) addresses these challenges through two subtasks using the multi-dialect Tamil speech corpus (Bharathi et al., 2025).

Subtask 1 requires classifying Tamil speech into four dialect categories, while Subtask 2 requires transcribing dialectal Tamil speech to text, evaluated by Word Error Rate (WER). Our approach evaluates multiple pre-trained speech models across both subtasks. For classification, we compare MMS-1b-all (Pratap et al., 2023) (1B parameters, 1,100+ languages) against wav2vec2-large-xlsr-53 (Conneau et al., 2021) (300M, 53 languages). For ASR, we compare a Tamil-specific Whisper variant (763M) against the general-purpose Whisper-large-v3 (Radford et al., 2023) (1.5B). Our experiments demonstrate that domain-specific pre-training consistently outperforms model scale across both tasks.[1]

2 Related Work

Speech processing for Indian languages has gained momentum through shared tasks and community efforts. Bharathi et al. (2022) organized the first shared task on speech recognition for vulnerable individuals in Tamil, establishing benchmarks for Tamil ASR. The multi-dialect Tamil speech corpus (Bharathi et al., 2025) used in this shared task provides a foundation for studying dialectal variation in Tamil speech.

Self-supervised speech models have transformed the field. Wav2Vec2 (Baevski et al., 2020) learns speech representations from unlabeled audio through contrastive learning. Its multilingual extension, XLSR-53 (Conneau et al., 2021), covers 53 languages. MMS (Pratap et al., 2023) scales further to 1,100+ languages with 1B parameters. For ASR, Whisper (Radford et al., 2023) achieves strong multilingual performance through weakly supervised pre-training on 680,000 hours of audio. Community-driven efforts have produced Tamil-specific Whisper variants fine-tuned on Tamil speech corpora, which we leverage in this work.

Focal Loss (Lin et al., 2017) has been successfully applied to speech classification tasks with class imbalance. Prior work on Dravidian language processing (Jada et al., 2021; Sai Kumar et al., 2021; Premjith et al., 2022) has shown

[1]The code for this work is available at: https://github.com/Arunaggiri-Pandian/Dialect_Based_Speech_Recognition_and_Classification_in_Tamil

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 186–190
July 3, 2026 ©2026 Association for Computational Linguistics

Dialect	Train	Dev	Test
Northern	1,525	171	189
Southern	1,186	241	212
Western	944	182	116
Central	716	168	62
Total	**4,371**	**762**	**579**

Table 1: Dataset distribution across four Tamil dialects.

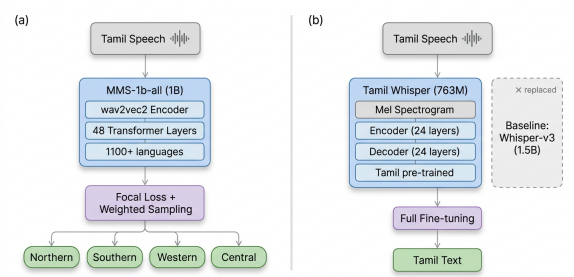


Figure 1: System architecture. (a) Dialect classification via MMS-1b-all. (b) Speech recognition via Tamil Whisper, which outperforms the 2x larger Whisper-large-v3 by 8 WER points.

that language-specific approaches consistently outperform general multilingual methods across text, speech, and multimodal tasks.

3 Methodology

3.1 Dataset

The dataset (Bharathi et al., 2025) contains Tamil speech samples across four regional dialects. Table 1 shows the distribution. Audio files are mono-channel WAV at 16 kHz sampling rate, with Tamil script transcriptions.

3.2 Subtask 1: Dialect Classification

We build a classification pipeline on top of self-supervised speech encoders. Given an input waveform, the encoder produces frame-level representations that are mean-pooled over time to obtain a fixed-dimensional utterance embedding $\mathbf{h} \in R^d$. This embedding is passed through a two-layer classification head:

$$\mathbf{y} = W_2 \cdot \text{ReLU}(W_1 \cdot \text{Dropout}(\mathbf{h}) + \mathbf{b}_1) + \mathbf{b}_2 \quad (1)$$

where $W_1 \in R^{d \times d/2}$ and $W_2 \in R^{d/2 \times 4}$. We evaluate two encoders:

- **MMS-1b-all** (Pratap et al., 2023): 1B parameters (d=1280), pre-trained on 1,100+ languages via self-supervised learning.
- **wav2vec2-large-xlsr-53** (Conneau et al., 2021): 300M parameters (d=1024), pre-trained on 53 languages.

Both encoders are fine-tuned end-to-end with Focal Loss (γ=2.0), WeightedRandomSampler for class balance, dropout (p=0.2), and early stopping on dev Macro F1 with patience of 5 epochs. We set γ=2.0 following Lin et al. (2017)'s recommended default, which down-weights well-classified dialects while focusing learning on harder dialect boundaries without overly aggressive focusing that higher gamma values can cause on small datasets. Figure 1 illustrates our system architecture for both subtasks.

Parameter	Classification	ASR
Learning rate	5e-6	3e-5
Batch size	16	16
Grad. accumulation	1	4
Loss	Focal (γ=2)	Cross-entropy
Precision	FP32	BF16
Early stop patience	5	10
Seed	42	42

Table 2: Training configuration for both subtasks.

3.3 Subtask 2: Speech Recognition

For ASR, we fine-tune Whisper encoder-decoder models using sequence-to-sequence training. The encoder processes mel-spectrogram features, and the decoder autoregressively generates Tamil script tokens. We compare two models:

- **Tamil Whisper medium**: 763M parameters, a community fine-tuned variant of Whisper-medium pre-trained on Tamil speech corpora.
- **Whisper-large-v3** (Radford et al., 2023): 1.5B parameters, the largest general-purpose Whisper model.

We fine-tune with BF16 mixed precision, gradient accumulation (effective batch size 64), AdamW optimizer (lr=3e-5), and 500 warmup steps. Label sequences exceeding 448 tokens are truncated. Table 2 summarizes configurations for both subtasks.

4 Results

4.1 Dialect Classification

Table 3 shows classification results. MMS-1b-all achieves the best dev F1 of 83.04%, outperforming wav2vec2-xlsr-53 by 17 points. Table 4 shows per-dialect F1 scores. Central dialect is easiest to classify (94.2% F1) while Northern is hardest (77.9%). Figure 2 shows the confusion matrix: Central is

Model	Dev MF1	Epoch
MMS-1b-all (tuned)	**83.04**	**7**
MMS-1b-all (initial)	79.87	5
wav2vec2-xlsr-53 (tuned)	66.38	8
wav2vec2-xlsr-53 (baseline)	57.48	10

Table 3: Dialect classification results (dev Macro F1 %).

Dialect	Precision	Recall	F1	Samples
Northern	69.4	88.9	77.9	171
Southern	81.1	76.8	78.9	241
Western	87.3	75.8	81.2	182
Central	97.5	91.1	94.2	168
Macro	**83.8**	**83.1**	**83.0**	**762**

Table 4: Per-dialect classification metrics (%) for MMS-1b-all.

cleanly separated while Northern is most often confused with Southern, reflecting their geographic and phonological proximity.

Figure 3 shows the training dynamics: MMS-1b-all reaches peak dev F1 of 83.04% at epoch 7 while train F1 continues rising past 95%, indicating rapid adaptation with some overfitting.

On the official test set, our system ranks 5th out of 11 teams with 43.09% Macro F1. Analysis of the test labels reveals a distribution shift across dialects: Southern (80.5%) and Northern (57.8%) remain stable, while Central and Western show sensitivity to the changed test distribution, indicating opportunities for more robust dialect-invariant features. Full results are in the task overview paper (Bharathi et al., 2026).

4.2 Speech Recognition

Table 5 shows ASR results. Tamil Whisper achieves 53.72% WER, outperforming Whisper-large-v3 by 8 points despite having half the parameters. Figure 4 visualizes this comparison.

Table 6 shows per-dialect WER on the dev set. A 40-point gap separates Central (28%) from Northern (69%).

On the official test set, our system achieves 49.75% WER, ranking 1st out of 13 teams. Table 7 shows per-dialect test WER. Central WER increases from 28% (dev) to 47% (test) while Northern improves from 69% to 52%, compressing all dialects into a 6-point range (Figure 5).

5 Analysis

Domain-specific pre-training is the decisive factor: Tamil Whisper (763M) outperforms Whisper-large-v3 (1.5B) by 8 WER points, and MMS-1b-all outperforms wav2vec2-xlsr-53 by 17 F1 points. Full fine-tuning outperforms LoRA by 9 WER points, and SpecAugment hurts (70.07% vs 61.65%) at this data scale. The classification dev-to-test gap (83% to 43%) reflects speaker distribution shift between splits. Central Tamil achieves the lowest WER (28% dev) due to phonological proximity to standard Tamil, while the narrower test gap (6 vs 40 points) suggests more extreme dialectal variation in the dev split.

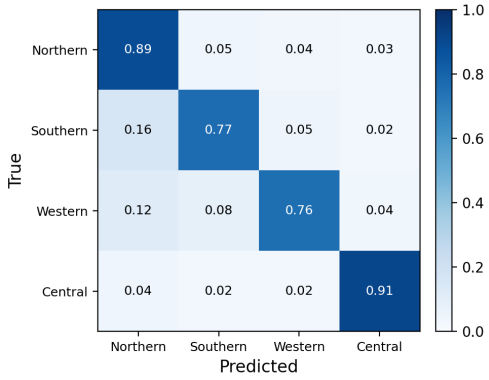


Figure 2: Normalized confusion matrix for MMS-1b-all. Central is cleanly separated; Northern is most confused with Southern.

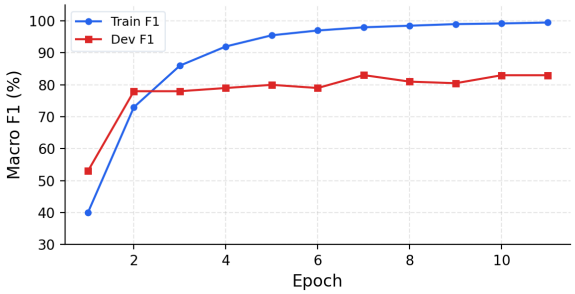


Figure 3: Dialect classification training curves for MMS-1b-all. Dev F1 peaks at epoch 7 while train F1 continues rising.

6 Conclusion

We presented a system for Tamil dialect speech processing achieving Rank 1 in ASR (49.75% WER, 13 teams). Key findings: (1) domain-specific pre-training outperforms scale (Tamil Whisper 763M beats Whisper-v3 1.5B by 8 WER points); (2) full fine-tuning outperforms LoRA by 9 points; (3) SpecAugment hurts on small datasets; (4) per-dialect WER compresses from 40 points (dev) to 6 (test).

Model	Params	WER
Tamil Whisper (full FT)	**763M**	**53.72**
Tamil Whisper (LoRA)	763M	62.93
Tamil Whisper (zero-shot)	763M	86.39
Whisper-large-v3 (best)	1.5B	61.65
Whisper-large-v3 (+SpecAug)	1.5B	70.07

Table 5: ASR results (dev WER %, lower is better).

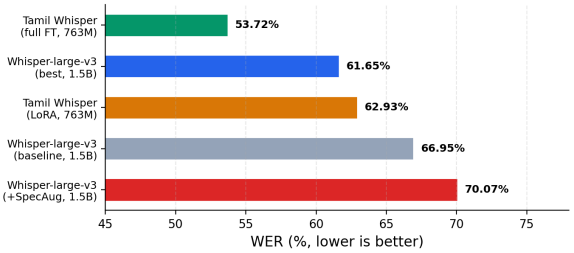


Figure 4: ASR model comparison. Tamil Whisper (763M) outperforms the twice-larger Whisper-large-v3 (1.5B) by 8 WER points.

Limitations

Dialect-adaptive decoding strategies that leverage the classification model to condition ASR output could yield further improvements. Multi-task learning jointly optimizing dialect classification and transcription is another avenue worth investigating. The significant dev-to-test gap in classification (83% to 43%) suggests potential distribution differences that merit closer examination. Incorporating dialect-specific language models may also help improve recognition for underperforming dialects.

Dialect	WER	CER	Samples
Central	28.07	7.31	168
Western	48.55	16.82	182
Southern	52.93	18.09	241
Northern	68.67	32.28	171
Overall	**53.72**	**21.61**	**762**

Table 6: Per-dialect WER and CER (%) on the dev set.

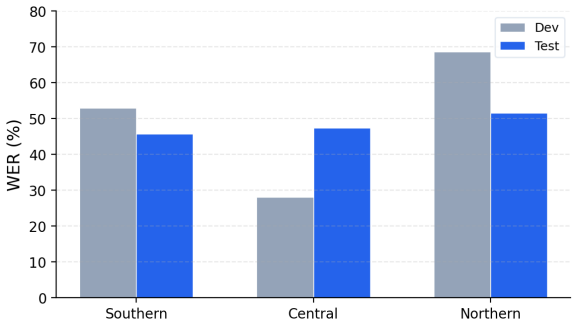


Figure 5: Per-dialect WER: dev (wide spread) vs test (6-point cluster).

Dialect	WER	CER	Samples
Southern	45.61	14.37	212
Central	47.36	13.65	62
Northern	51.50	16.42	189
Overall	**49.75**	**15.59**	**463**

Table 7: Per-dialect test WER and CER (%). Western excluded.

References

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems (NeurIPS)*.

B. Bharathi, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, S. Saranya, and S. Suhasini. 2026. Findings in Tamil Dialect Speech Recognition and Classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

B. Bharathi, Bharathi Raja Chakravarthi, Subalalitha Cn, N. Sripriya, Arunaggiri Pandian Karunanidhi, and Swetha Valli. 2022. Findings of the shared task on speech recognition for vulnerable individuals in Tamil. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*. Association for Computational Linguistics.

B. Bharathi, S. Saranya, P. Vijayalakshmi, and T. Nagarajan. 2025. Multi-dialect speech corpus creation for enhancing Tamil automatic speech recognition. *Circuits, Systems, and Signal Processing*, pages 1–19.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2021. Unsupervised cross-lingual representation learning for speech processing. In *Proceedings of Interspeech*.

Pawan Kalyan Jada, D. Sashidhar Reddy, Konthala Yasaswini, Arunaggiri Pandian Karunanidhi, Prabakaran Chandran, Anbukkarasi Sampath, and Sathiyaraj Thangasamy. 2021. Transformer based sentiment analysis in Dravidian languages. In *Proceedings of FIRE 2021 (Working Notes)*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tober, Changhan Babu, Sayahna Kunber, Ali Elkahky, Zhaoheng Xue, Arya Fazel-Zarandi, Alexei Baevski, Yossi Adi, Wei-Ning Zhang, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2023. Scaling speech

technology to 1,000+ languages. *arXiv preprint arXiv:2305.13516*.

B. Premjith, Bharathi Raja Chakravarthi, Malliga Subramanian, B. Bharathi, K.P. Soman, V. Dhanalakshmi, K. Sreelakshmi, Arunaggiri Pandian Karunanidhi, and Prasanna Kumaresan. 2022. Findings of the shared task on multimodal sentiment analysis and troll meme classification in Dravidian languages. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. *Proceedings of the 40th International Conference on Machine Learning (ICML)*.

T.S. Sai Kumar, K. Arunaggiri Pandian Karunanidhi, S. Thabasum Aara, and K. Nagendra Pandian. 2021. A reliable technique for sentiment analysis on tweets via machine learning and BERT. In *2021 Asian Conference on Innovation in Technology (ASIANCON)*. IEEE.

Cuet Yet Another Baseline @ DravidianLangTech 2026: Shared Task on Prompt Recovery for LLM in Telugu

Rotna Dipika Debnath, Shahrin Afroz Hoque Ruhi, Ayesha Labiba, Arpita Mallik, Hasan Murad
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology, Bangladesh
{u2104002, u2104015}@student.cuet.ac.bd,
{u2104008, u2004023}@student.cuet.ac.bd, hasanmurad@cuet.ac.bd

Abstract

Prompt recovery in large language models (LLMs) is the task of inferring the communicative intent and stylistic framing of the original instruction from model-generated output. This task is especially challenging for low-resource Dravidian languages such as Telugu, where agglutinative morphology, register variation, and scarce annotated data complicate stylistic modelling. In this paper, we present our system for the Shared Task on Prompt Recovery for LLM in Telugu at DravidianLangTech @ ACL 2026, which aims to classify Telugu transcript excerpts into nine communicative style categories: Formal, Informal, Optimistic, Pessimistic, Humorous, Serious, Inspiring, Authoritative, and Persuasive.We have implemented a transformer-based approach using ai4bharat/IndicBERTv2-MLM-only, MuRIL-base and Telugu-BERT for Telugu communicative style classification. Our system fine-tunes the pretrained Indic language training samples to capture stylistic patterns in Telugu transcripts. Our approach achieved a macro F1 score of 0.2993 on the evaluation set, demonstrating the potential of Indic-focused pretrained models for stylistic analysis in low-resource language settings.Controlled ablations reveal that label smoothing benefits stronger Indic backbones but degrades weaker ones, and that surface linguistic feature augmentation does not complement rich contextual representations on small datasets.

1 Introduction

Large language models (LLMs) have become the dominant paradigm for natural language processing, with carefully designed prompts being the primary mechanism for controlling model behaviour (Schulhoff et al., 2024).

Recovering the stylistic intent of a prompt from the generated output is critical for interpretability, safety, and controlled generation (Brown et al., 2020; Wei et al., 2022). For Telugu, a Dravidian language with agglutinative morphology and over 80 million native speakers, this problem is further complicated by limited labelled resources and underrepresentation in major multilingual pretraining corpora.

The DravidianLangTech @ ACL 2026 conference(Chakravarthi et al., 2026) has introduced a dataset of 3,601 Telugu transcripts rewritten in one of nine styles by an LLM. Given the rewritten text, a system must predict the style label. We have addressed this task by fine-tuning IndicBERT v2 (Doddapaneni et al., 2023), a multi-stage pretrained encoder covering 23 Indian languages, augmented with label smoothing regularisation. We have compared it against Telugu-BERT(Joshi et al., 2022) and a MuRIL-based hybrid with handcrafted linguistic features(Khanuja et al., 2021).

The main contributions of this work have been:

- Fine-tuning of ai4bharat/IndicBERTv2-MLM-only with label smoothing for Telugu style classification, ranking 1st on the shared-task leaderboard.
- Controlled ablations of label smoothing across three backbones and linguistic feature augmentation on IndicBERT v2, revealing when each technique helps or hurts.
- A reproducible pipeline with fixed random seeding and early stopping for stable convergence.

Further details of the implementation can be accessed through the GitHub repository: [1].

[1] https://github.com/prism-rkive/prompt_recovery_indic

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 191–195
July 3, 2026 ©2026 Association for Computational Linguistics

2 Related Work

IndicBERT (Kakwani et al., 2020) introduced an ALBERT-based compact multilingual encoder for 12 Indian languages. IndicBERT v2 (Doddapaneni et al., 2023) extended this with a three-stage pretraining pipeline (MLM, TLM, task-adaptive) using the XLM-RoBERTa architecture and SentencePiece tokenisation across 23 languages, yielding large gains on downstream Indic NLP tasks. MuRIL (Khanuja et al., 2021) targets transliteration and code-mixing phenomena across 17 Indian languages. Telugu-BERT (Joshi et al., 2022) demonstrated that monolingual Telugu pretraining on 6.2 GB of text outperforms multilingual baselines on Telugu classification tasks.

Fine-tuned BERT representations outperform traditional stylometric pipelines for authorship and style attribution (Fabien et al., 2020). Integrating handcrafted linguistic features with Transformer embeddings via early fusion further improves performance in low-resource settings (Zhao et al., 2021; Nitu et al., 2024). Label smoothing (Müller et al., 2019) improves calibration in multi-class settings by preventing overconfident predictions, which is particularly beneficial when class boundaries are subtle and nine-way confusion is likely. Focal Loss (Lin et al., 2017) and SMOTE (Chawla et al., 2002) address class imbalance but provide no benefit when the dataset is already balanced (Section 3).

3 Data

We have utilized the dataset provided under the Shared Task on Prompt Recovery for LLM in Telugu DravidianLangTech @ ACL 2026 (Chakravarthi et al., 2026). The dataset supports style transfer and controlled text generation, where each instance pairs an original Telugu transcript with a stylistically rewritten version of the same content.

The dataset is provided in Excel (.xlsx) format with four columns: ID, ORIGINAL TRANSCRIPTS, CHANGE STYLE, and STYLE. The ORIGINAL TRANSCRIPTS column contains the source text, CHANGE STYLE provides the style-modified input, and STYLE indicates the target label.

The dataset is split into 3,000 training, 300 development, and 301 test instances, as shown in Table 1. The training split is nearly balanced across all nine classes (321–347 samples,10.7%-11.6%), as shown in Table 2, confirming that class imbalance techniques such as Focal Loss and SMOTE provide no structural benefit.

Language	Train	Development	Test	Total
Telugu	3,000	300	301	3,601

Table 1: Dataset statistics for the Prompt Recovery for LLM in Telugu shared task.

Style Class	Train	%	Class Weight
Pessimistic	347	11.6	0.96
Humorous	344	11.5	0.98
Authoritative	338	11.3	0.99
Persuasive	336	11.2	1.02
Inspiring	332	11.1	0.99
Optimistic	331	11.0	0.99
Formal	327	10.9	1.01
Serious	324	10.8	1.01
Informal	321	10.7	1.02
Total	3,000	100.0	0.96–1.02

Table 2: Style class distribution in the training split. Class weights are computed as $w_c = \bar{n}/n_c$ where $\bar{n}$ is the mean class count.

4 Methodology

4.1 Data Preprocessing

In terms of data preprocessing, labels have been normalized by trimming whitespace and converting to title case, and samples with missing text or labels have been discarded. Each label has been mapped to one of nine style indices: Formal (0), Informal (1), Optimistic (2), Pessimistic (3), Humorous (4), Serious (5), Inspiring (6), Authoritative (7), and Persuasive (8). A fixed random seed (42) has been set for Python, NumPy, and PyTorch (including CUDA) with deterministic CuDNN for reproducibility.

4.2 Overview of Experimented Models

4.2.1 IndicBERT v2

In this approach, We have fine-tuned ai4bharat/IndicBERTv2-MLM-only(278M parameters, XLM-RoBERTa architecture) (Doddapaneni et al., 2023) for sequence classification using HuggingFace Transformers. A randomly initialised linear head projects the 768-dimensional

[CLS] token embedding from the final hidden layer to nine output logits. Input texts from the CHANGE STYLE column are tokenised with SentencePiece (max_length= 256, padding and truncation enabled); the 200K-subword vocabulary covers 23 Indian languages, providing robust segmentation for agglutinative Telugu morphology. The overall architecture is illustrated in Figure 1.

The model is optimised with AdamW (lr $= 2\times10^{-5}$, weight_decay $= 0.01$, batch_size $= 16$) under a linear learning rate decay schedule with no warmup over 1,880 total steps (188 batches $\times$ 10 epochs). Gradients are clipped to max_norm $= 1.0$.

The loss function is CrossEntropyLoss with label smoothing $\varepsilon = 0.1$, which replaces the hard one-hot target for the true class y with a soft distribution:

$$\tilde{q}_k = \begin{cases} 1 - \varepsilon + \dfrac{\varepsilon}{K} & k = y, \\ \dfrac{\varepsilon}{K} & k \neq y, \end{cases} \tag{1}$$

where $K = 9$. Following Müller et al. (2019), this prevents overconfident predictions — particularly important here because adjacent style classes (e.g., Formal, Serious, Authoritative) share substantial lexical and syntactic overlap. Dropout $= 0.3$ is applied to the classification head as additional regularisation. The label smoothing coefficient $\varepsilon = 0.1$ and classification-head dropout of 0.3 follow the fine-tuning recommendations of Müller et al. (2019) and Doddapaneni et al. (2023) respectively; a full grid search was infeasible given the shared-task timeline.

Training runs for at most 10 epochs, saving the checkpoint with the highest validation Macro F1. Training stops early if no improvement is seen for three consecutive epochs (patience $= 3$); the best checkpoint was obtained at epoch 4. At inference, predicted labels are produced by $\hat{y} = \arg\max_k \text{logit}_k$ and mapped back to style-name strings for submission.

4.2.2 Other Experimented models

We have explored various other methods such as fine-tuning l3cube-pune/telugu-bert (238M parameters, BERT-large with 24 layers) using AdamW with cosine learning rate decay, where the official training and validation splits were merged and re-partitioned into a 90/10 train–validation split with a three-seed ensemble to improve stability. We have also experimented with augmenting google/muril-base-cased (237M parameters) with a 44-dimensional handcrafted Telugu linguistic feature vector, fused with the [CLS] embedding through a multilayer perceptron (MLP) before the final classification layer (Qi et al., 2020).To ensure a fair comparison, we subsequently applied identical label smoothing regularisation ($\varepsilon = 0.1$) to both Telugu-BERT and MuRIL; even under matched training conditions both were outperformed by IndicBERT v2 (Table 3).

5 Results and Analysis

5.1 Comparative Analysis and Ablation

Table 3 presents test-set results for all systems. Our primary IndicBERT v2 system achieves a test Macro F1 of 0.2993 (Precision: 0.3146, Recall: 0.3043, Accuracy: 0.2890). Expanding max_length from 256 to 512 yields a marginal gain (+0.0038 F1), confirming that truncation is a contributing but not primary bottleneck. Telugu-BERT's F1 decreases with label smoothing (0.2160 $\rightarrow$ 0.1713), confirming the calibration benefit requires a strong backbone to amplify. Adding linguistic features to IndicBERT v2 decreases F1 to 0.2528, with Serious collapsing to F1 $= 0.000$, suggesting the fusion MLP overfits on surface proxies when the backbone already encodes rich contextual representations.

System	LS	Test Macro Average			
		P	R	F1	Acc.
IndicBERT v2 †	✓	0.3146	0.3043	**0.2993**	0.2890
IndicBERT v2 (512 tok.) ‡	✓	0.3046	0.3072	0.3031	0.2924
IndicBERT v2 + Ling. Feat. ‡	✓	0.2448	0.2710	0.2528	0.2558
Telugu-BERT	×	0.2354	0.2634	0.2160	0.2700
Telugu-BERT (w/ LS)	✓	0.1885	0.2289	0.1713	0.2159
MuRIL + Ling. Feat. (w/ LS)	✓	0.2406	0.2356	0.2285	0.2193

Table 3: Test-set and ablation results. † = primary submission; ‡ = post-hoc analysis, not submitted; LS = label smoothing ($\varepsilon = 0.1$). Macro F1 is the official metric.

5.2 Per-class Performance

Table 4 shows per-class results. Optimistic achieves the highest F1 (0.4848) while Serious is the weakest (F1 $= 0.0845$), likely due to its dependence on sustained register consistency across passages truncated at max_length= 256.

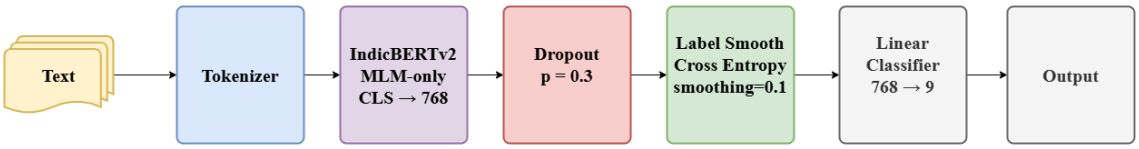


Figure 1: Architecture of the IndicBERT v2 system for Telugu style classification.

Style	P	R	F1	Support
Formal	0.1528	0.2895	0.2000	38
Informal	0.3514	0.3714	0.3611	35
Optimistic	0.4848	0.4848	0.4848	33
Pessimistic	0.4000	0.3636	0.3810	22
Humorous	0.2424	0.2581	0.2500	31
Serious	0.1250	0.0638	0.0845	47
Inspiring	0.5000	0.2432	0.3273	37
Authoritative	0.2500	0.2000	0.2222	30
Persuasive	0.3250	0.4643	0.3824	28
Macro avg	**0.3146**	**0.3043**	**0.2993**	**301**

Table 4: Per-class test-set results for our IndicBERT v2 system.

5.3 Error Analysis

Figure 2 shows the test-set confusion matrix. Formal is over-predicted (67 of 301), acting as a catch-all for neutral-register text. Serious is the weakest class (recall = 0.0638) and Inspiring shows high precision (0.50) but low recall (0.2432), indicating the model is conservative about predicting it. Three dominant confusion patterns, all sharing the same root cause, are detailed in Table 5: stylistic signals that disambiguate adjacent categories are discourse-level and passage-final, not token-level and passage-initial.

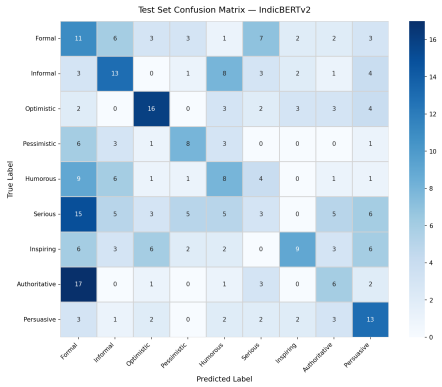


Figure 2: Confusion Matrix of the IndicBERT v2 Model.

Sample	True	Pred.	Reason
ప్రాచీన భారతదేశ చరిత్ర: అధ్యయన విధానం మరియు ఆధారాలు (Ancient Indian History: Methods and Sources of Study)	Serious	Formal	Grave register emerges past truncation boundary
అందరికీ నమస్కారం అండి (Greetings to all)	Inspiring	Optimistic	Aspirational moral framing is in truncated portion
జలియన్ వాలాబాగ్: ఉమర్ అలీషా రచనలో చారిత్రక విశ్లేషణ (Jallianwala Bagh: Historical Analysis in Umar Alisha's Writing)	Authoritative	Formal	Expert voice builds progressively beyond context window

Table 5: Representative misclassification examples (17/47, 11/37, and 12/30 instances respectively).

6 Conclusion

We present a Telugu communicative style classification system for the DravidianLangTech @ ACL 2026 shared task, achieving a Macro F1 of 0.2993, with ablations showing that label smoothing and backbone quality interact such that stronger Indic pretraining amplifies calibration benefits while weaker backbones are destabilised by soft targets, linguistic feature augmentation via MLP fusion consistently underperforms the plain encoder due to overfitting on small datasets, and error analysis reveals the primary failure mode is discourse-level, where passage-final stylistic signals distinguishing adjacent categories are lost to truncation, motivating future work on cross-attention feature fusion, ensemble methods, dataset expansion, and sliding-window attention for long-document truncation.

Limitations

The 3,000-sample training set limits the model's ability to learn fine-grained stylistic distinctions, particularly for categories with subtle register differences such as Formal, Serious, and Authoritative. Expanding max_length from 256 to 512 yields a modest improvement (Macro F1: 0.2993 → 0.3031) but does not resolve the low performance on Serious (F1 = 0.0769), suggesting

that truncation alone is not the primary bottleneck and that discourse-level modelling beyond token-window extension is needed. Our study is restricted to BERT-style encoder models; decoder-only or longer-context architectures remain unexplored due to computational constraints. Finally, we report point estimates without bootstrapped confidence intervals or significance tests; with 301 test samples, such intervals would be a valuable addition in future work.

Ethical Considerations

We have done our research, maintaining all ethical standards in a responsible manner. Our work classifies pre-existing, synthetically generated Telugu text into predefined style categories. The dataset contains no personally identifiable information, and our models do not generate new content. Through our work, we intend to support the development of technologies that elevate the understanding of low-resource languages.

Acknowledgements

The authors used GitHub Copilot for boilerplate code completion during implementation. All scientific contributions, analysis, and writing are the authors' own.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, and 1 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Bharathi Raja Chakravarthi and 1 others. 2026. Overview of the shared task on prompt recovery for LLM in Telugu. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages (DravidianLangTech)*. Association for Computational Linguistics.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Sumanth Doddapaneni, Gowtham Kohli, Aman Doddapaneni, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2023. IndicBERTv2: Multi-stage pre-training for Indic NLP. *arXiv preprint arXiv:2212.05236*.

Maël Fabien, Esaú Villatoro-Tello, Petr Motlicek, and Shantipriya Parida. 2020. BertAA: BERT fine-tuning for authorship attribution. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*.

Raviraj Joshi, Purva Nankambakkam, Devang Limaye, Pradnya Thakkar, and Chinmay Kanchan. 2022. L3Cube-HindBERT and DevBERT: Fine-tuned BERT language models for Hindi and Devanagari. *arXiv preprint arXiv:2211.11418*.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, N. Chandra Shekhar Gokul, Avik Iyer, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Bharadwaj, and Partha Talukdar. 2021. MuRIL: Multilingual representations for Indian languages. *arXiv preprint arXiv:2103.10730*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988.

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. 2019. When does label smoothing help? In *Advances in Neural Information Processing Systems*, volume 32.

Mihaela Nitu, Cornelia Caragea, and Doina Caragea. 2024. Authorship attribution using stylometric features and transformer models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108.

Sander Schulhoff, Michael Ilie, Nishant Bhatt, and 1 others. 2024. The prompt report: A systematic survey of prompting techniques. *arXiv preprint arXiv:2406.06608*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35.

Haining Zhao, Oren Halvani, Lukas Miculivicius, and Benno Stein. 2021. Linguistic features meet transformer encoders for authorship attribution. *arXiv preprint arXiv:2111.14538*.

CUET-2567@DravidianLangTech-ACL 2026: Multimodal Stance and Target Identification in Dravidian Political Memes

Arka Dutta, Anindya Majumder, Adnan Faisal, Hasan Murad
Department of Computer Science and Engineering,
Chittagong University of Engineering and Technology, Bangladesh
{u2204025, u2204067, u2004002}@student.cuet.ac.bd,hasanmurad@cuet.ac.bd

Abstract

In Dravidian languages, political memes progressively shape public opinion and political discourse, influencing digital conversations and public narratives. Our paper proposes a multilevel multimodal framework for political meme classification in Tamil and Malayalam as part of the Multi Level Political Meme Classification-DravidianLangTech@ACL 2026 shared task. The task has involved two levels: Level 1 has identified whether a meme expresses Troll/Oppose or Support/Praise, while Level 2 has determined the specific target category (Individual, Party, or Intersection). We have evaluated unimodal and multimodal architectures to analyze the impact of textual and visual representation. Experimental results have highlighted the importance of a multimodal approach over unimodal approaches. This work confirms the effectiveness of combining image and text features in meme understanding. Among the evaluated models, the mBERT+ViT architecture has achieved the best overall performance across both languages and classification levels. According to the evaluation of shared task we achieved average F1 score of 0.72 securing the 2nd rank in Malayalam task and F1 score of 0.76 in Tamil task securing the 6th rank. However after our experimental evaluation we got best average F1 score of 0.62 for Tamil and 0.49 for Malayalam. Despite moderate results, challenges have remained mainly due to the dataset size, class imbalance, and noisy text extraction from images.

1 Introduction

As internet accessibility has increased, memes have become influential units of cultural transmission (Dawkins, 1976), often combining text and images to shape political discourse (Afridi et al., 2021). However, political meme classification—especially for Tamil and Malayalam—remains underexplored, highlighting the need for effective multimodal approaches (Galipeau, 2023).

The Multi-Level Political Meme Classification - DravidianLangTech@ACL 2026 Shared Task addresses the challenge of classifying political memes in Dravidian languages, requiring models to perform analysis across multiple levels, including sentiment identification (Support/Praise vs. Troll/Oppose) and fine-grained target categorization (Individual, Party, or Intersection

In our task on the Tamil meme dataset, we have explored multiple multimodal and unimodal combinations for multilevel classification. Among them, multilingual BERT (mBERT) for textual features combined with Vision Transformer (ViT) for visual features achieved the best performance. Specifically, Level 1 has achieved an F1-score of 0.68, and Level 2 has achieved an F1-score of 0.53. This combination has outperformed alternative setups, including CLIP + XLM-RoBERTa and IndicBERT + ResNet, indicating that mBERT captures textual nuances and ViT extracts complementary visual features, with the multimodal combination yielding more balanced performance than single-modality models. Similarly, on the Malayalam dataset, the fusion of BERT and ViT has outperformed combinations such as XLM-RoBERTa + CLIP and IndicBERT + ResNet, achieving an F1-score of 0.53 for Level 1 and 0.42 for Level 2. These results have demonstrated the effectiveness of multimodal fusion for multilevel political meme classification across languages. The core contributions of this work are summarized as follows:

- We have proposed a multilevel multimodal framework for Tamil and Malayalam political meme classification (Level 1 and Level 2).
- We have demonstrated that mBERT + ViT consistently have outperformed other multimodal combinations through systematic comparison.
- We have implemented a preprocessing pipeline that includes text transliteration and

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 196–200
July 3, 2026 ©2026 Association for Computational Linguistics

image enhancement techniques to improve overall data quality

For a comprehensive guide on the implementation process and to access the complete codebase, please visit the GitHub repository: CUET-2567 Multi-Level Political Meme Classification

2 Related Work

Political meme classification has emerged as a significant research area, as memes inherently integrate textual and visual cues that require multimodal deep learning frameworks for comprehensive semantic interpretation.

Prior work highlights the superiority of multimodal approaches, including Chinese BERT+VGG16 for misogynistic memes (Faisal et al., 2025), the Hateful Memes benchmark exposing unimodal limitations (Kiela et al., 2020), ViT+BERT with self-attentive fusion for multitask classification (Hossain et al., 2026), and BERT+GGNN for effective sarcasm detection (Qin et al., 2025).

Vision–language models such as OSCAR+RF (Chen and Pan, 2022), transformer-based Tamil troll detection (Hariprasad et al., 2022), and Bangla BERT+ViT on the MAC dataset (Alam et al., 2024) further demonstrate the effectiveness of multimodal fusion over unimodal approaches in meme classification.Motivated by prior multimodal meme analysis studies, our system adopts a late-fusion transformer-based framework (mBERT+ViT) tailored to the multi-level political meme classification task in Tamil and Malayalam.

3 Data Description

The datasets consist of Tamil and Malayalam political memes, each containing an image and corresponding textual labels. The Tamil dataset includes 1003 memes (Train: 802, Validation: 100, Test: 101, Total: 1003), while the Malayalam dataset includes 600 memes (Train: 500, Validation: 50, Test: 50, Total: 600). The training dataset has been used to learn model parameters, the validation dataset has been used for model selection and hyperparameter tuning during training, and the test dataset has been used only for final evaluation. The data distribution is shown in Table 1.

Level	Category	Malayalam	Tamil	Total
Level 1	Troll/Oppose	477	691	1168
	Support/Praise	23	112	135
Level 2	Individual Person	327	633	960
	Party	120	170	290
	Intersection	53	0	53
Total		1000	1606	2606

Table 1: Dataset Statistics for Tamil and Malayalam Political Memes

4 Methodology

4.1 Problem Formulation

The task has been formulated as a multimodal multi-task classification problem. Given a political meme m that has been composed of a textual component t and a visual component i, the objective has been to perform hierarchical classification at two levels:

- **Level 1:** Support vs. Troll/Oppose
- **Level 2:** Person vs. Party vs. Intersection

Let $t \in R^m$ denote the textual representation of features that have been extracted from the meme text, and $i \in R^m$ denote the visual representation of features that have been extracted from the meme image. The goal has been to learn a mapping function:

$$f(t, i) \rightarrow \{y_1, y_2\} \quad (1)$$

As defined in Eq. 1, where $y_1 \in \{0, 1\}$ has represented the Level 1 label (0 = Support, 1 = Troll/Oppose), and $y_2 \in \{0, 1, 2\}$ has represented the Level 2 label (0 = Person, 1 = Party, 2 = Intersection). The model has leveraged multimodal fusion of textual and visual features to maximize multilevel classification performance.

4.2 Data Preprocessing

4.2.1 Text Preprocessing

For the textual modality, pretrained multilingual tokenizers have been employed with a maximum sequence length of 128 tokens. Unwanted symbols, punctuation, numbers, URLs, and emojis have been removed. Language-specific stopwords for Tamil and Malayalam have been eliminated to preserve semantically meaningful content. Contextual embeddings have been extracted using transformer-based encoder models.

4.2.2 Image Preprocessing

For the visual modality, images have been resized to 224×224 pixels and have been normalized using ImageNet mean and standard deviation statistics. The processed images are converted into tensor representations. Global visual features are extracted using pretrained backbone networks.

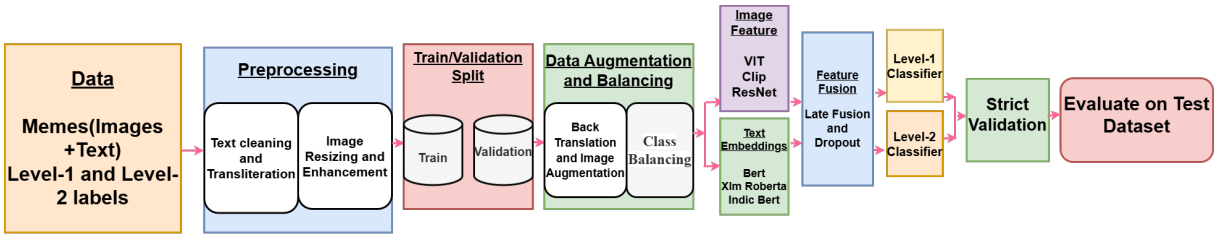


Figure 1: Multimodal Process Flow Framework for Detecting Political Memes

4.3 Data Augmentation

To improve generalization and address class imbalance, data augmentation has been applied to the image modality. Minority classes have been augmented in both Level 1 and Level 2 for Tamil and Malayalam datasets to reduce class disparity.

For the textual modality, a back-translation strategy has been employed using the *deep-translator* framework to generate semantically equivalent paraphrased samples.For the visual modality, a data augmentation strategy has been applied using controlled geometric and color transformation to generate visually varied yet semantically preserved image samples.

4.4 Multimodal Architecture

We implement a late-fusion multimodal architecture with separate pretrained text encoders (mBERT, XLM-RoBERTa, IndicBERT) and vision backbones (ViT, CLIP, ResNet-50), whose representations are concatenated and fed into a dropout layer followed by two parallel fully connected heads for Level-1 and Level-2 classification. As illustrated in Figure 1, the framework integrates preprocessing, augmentation, feature extraction, and fusion within a unified pipeline. Among all configurations, mBERT+ViT achieves the best performance on both Tamil and Malayalam datasets, using a maximum sequence length of 128 tokens and image resolution of 224×224.

4.5 Evaluation Metrics

Model performance was evaluated using Accuracy, Precision, Recall, and F1-score. Due to potential class imbalance, Macro-F1 was considered the primary evaluation metric. Performance was assessed separately for Level 1 and Level 2 classification tasks on both the Tamil and Malayalam datasets.

4.6 Parameter Setting

We have used AdamW as the optimizer and cross-entropy loss for training. ViT + mBERT uses a higher learning rate because both components are more stable during fine-tuning and require faster adaptation, whereas XLM-R and CLIP are more sensitive and need smaller learning rates to avoid overfitting or instability. Table 2 lists the hyperparameters used for XLM-RoBERTa + CLIP, mBERT + ViT, and IndicBERT + ResNet.

Model	Learning Rate	Batch Size	Epochs
ViT + mBERT	1e-4	16	10
XLM-R + CLIP	2e-5	16	10
IndicBERT + ResNet	2e-5	16	10

Table 2: Key Hyperparameters for Model Training

4.7 Result Analysis

This section presents the experimental results and comparative analysis of the proposed multimodal approaches. The models were analyzed based on their overall classification effectiveness across different tasks and datasets, with particular emphasis on Macro-F1 performance.

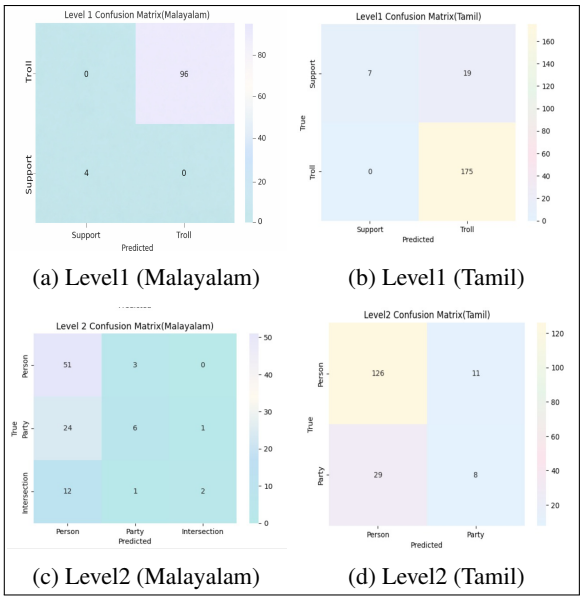
4.8 Comparative Analysis

We evaluate three multimodal configurations—ViT+mBERT, XLM-R+CLIP, and IndicBERT+ResNet—on Tamil and Malayalam datasets. As shown in Table 3, ViT+mBERT achieves the best performance across both levels, obtaining weighted F1 scores of 0.68 (Tamil) and 0.49 (Malayalam) at Level-1, and 0.53 (Tamil) and 0.42 (Malayalam) at Level-2. While Level-1 consistently yields higher scores, Level-2 remains more challenging due to fine-grained target distinctions. The results indicate that ViT+mBERT effectively captures complementary textual and visual patterns with stable generalization under strict validation (dropout = 0.3). The relatively lower performance in Malayalam and Level-2 primarily reflects dataset limitations, class imbalance, noisy OCR text, and intrinsic task complexity rather than model overfitting.

Figure 2 indicates strong but imbalanced Level-1 performance. For Malayalam, the model correctly classified 96 Troll and 4 Support instances, with

Lang	Model	P	R	F1
	Level-1			
Tamil	Text (mBERT)	0.68	0.53	0.50
Tamil	Image (ViT)	0.83	0.71	0.66
Tamil	Multi (mBERT+ViT)	**0.95**	**0.63**	**0.68**
Tamil	Multi (CLIP+XLM-R)	0.91	0.60	0.65
Tamil	Multi (IndicBERT+ResNet)	0.88	0.58	0.61
Malayalam	Text (mBERT)	0.47	0.46	0.48
Malayalam	Image (ViT)	0.46	0.49	0.47
Malayalam	Multi (mBERT+ViT)	**0.48**	**0.50**	**0.49**
Malayalam	Multi (CLIP+XLM-R)	0.47	0.49	0.48
Malayalam	Multi (IndicBERT+ResNet)	0.46	0.48	0.47
	Level-2			
Tamil	Text (mBERT)	0.39	0.50	0.44
Tamil	Image (ViT)	0.59	0.56	0.53
Tamil	Multi (mBERT+ViT)	**0.52**	**0.59**	**0.57**
Tamil	Multi (CLIP+XLM-R)	0.50	0.55	0.52
Tamil	Multi (IndicBERT+ResNet)	0.48	0.53	0.50
Malayalam	Text (mBERT)	0.35	0.37	0.35
Malayalam	Image (ViT)	0.52	0.46	0.42
Malayalam	Multi (mBERT+ViT)	**0.61**	**0.41**	**0.49**
Malayalam	Multi (CLIP+XLM-R)	0.49	0.39	0.36
Malayalam	Multi (IndicBERT+ResNet)	0.41	0.36	0.32

Table 3: Performance Comparison of models in Tamil and Malayalam Datasets



(a) Level1 (Malayalam) (b) Level1 (Tamil)

(c) Level2 (Malayalam) (d) Level2 (Tamil)

Figure 2: Confusion matrices for Level-1 and Level-2 classification in Tamil and Malayalam using mBERT and ViT.

no misclassification of Troll but a clear bias toward predicting Troll. For Tamil, it correctly identified 175 Troll and 7 Support, with 19 Support samples misclassified as Troll, again showing similar bias.

At Level-2 (Malayalam), the model correctly classified 51 Person, 6 Party, and 2 Intersection instances, but showed notable confusion, with many Party and Intersection samples misclassified as Person. At Level-2 (Tamil), the model correctly classified 126 Person and 8 Party instances, while misclassifying 29 Party as Person and 11 Person as Party, indicating a similar bias.

Overall, Level-1 demonstrates high accuracy, especially for the Troll class, whereas Level-2 remains more challenging due to class imbalance and fine-grained distinctions.

5 Error Analysis

The Level-1 results indicate that the models are quite effective at distinguishing between Troll and Support content in both Malayalam and Tamil, with relatively few misclassifications in each language. At Level-2, however, errors increase due to the more fine-grained class definitions. For Malayalam, most confusion occurs between posts about individuals, political parties, and intersectional content, while in Tamil the errors are mainly between individual and party-related posts. This pattern suggests that subtle contextual cues and overlapping semantic signals make fine-grained classification more difficult. Across these results, multimodal fusion shows limited additional benefit when one modality already captures the dominant signal, whereas improvements are more likely when both modalities contribute complementary contextual information. For future work, improvements in multimodal fusion and external knowledge integration can help reduce confusion arising from fine-grained class granularity and overlapping contextual cues.

6 Conclusion

In this work, we have presented a multi-level meme classification framework for Tamil and Malayalam political memes. We have conducted a comparative analysis of various unimodal and multimodal architectures for detecting political content in memes. The experimental results have demonstrated that multimodal fusion of textual and visual features have significantly improved classification performance, highlighting the importance of integrating both modalities. Among the evaluated models, the mBERT+ViT architecture has achieved the best overall performance across both Level 1 and Level 2 tasks. Future work should ensure better generalization by exploring more extensive and diverse datasets. Additionally, experimenting with larger vision language pre-trained models may enhance the performance.

7 Limitations

Despite moderate results, limitations persist due to small and imbalanced datasets, noisy OCR-based text extraction, and reliance on general-purpose pretrained encoders without task-specific linguistic adaptation, which may restrict the capture of nuanced cultural and contextual cues in regional political memes.

Acknowledgment

The authors would like to thank the organizers of DravidianLangTech@ACL 2026 for providing the competition platform and resources that facilitated this research. We also acknowledge the dataset introduced in the shared task overview paper (Rajiakodi et al., 2026), which formed the basis of our experiments.

Ethical Statement

All data processing and modeling followed ethical guidelines for handling sensitive, political content. The study aims to improve political memes detection while protecting users' rights and privacy. The goal is to improve automated moderation of political memes on online platforms. We have addressed any biases or limitations in the dataset to the best of our ability.

References

Tariq Afridi, Aftab Alam, Numan Khan, and Jawad Khan. 2021. *A Multimodal Memes Classification: A Survey and Open Research Issues*, pages 1451–1466.

Md Ashraful Alam, Jawad Hossain, Shawly Ahsan, and Mohammed Moshiul Hoque. 2024. Multimodal aggressive meme classification using bidirectional encoder representations from transformers. In *2024 27th International Conference on Computer and Information Technology (ICCIT)*, pages 3542–3547.

Yuyang Chen and Feng Pan. 2022. Multimodal detection of hateful memes by applying a vision-language pre-training model. *PLOS ONE*, 17(9):e0274300.

Richard Dawkins. 1976. *The Selfish Gene*. Oxford University Press, Oxford, U.K.

Adnan Faisal, Shiti Chowdhury, Momtazul Arefin Labib, and Hasan Murad. 2025. Team_Luminaries_0227@LT-EDI-2025: A transformer-based fusion approach to misogyny detection in Chinese memes. In *Proceedings of the 5th Conference on Language, Data and Knowledge: Fifth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 116–120, Naples, Italy. Unior Press.

Thomas Galipeau. 2023. The impact of political memes: A longitudinal field experiment. *Journal of Information Technology & Politics*, 20(4):437–453.

Shruthi Hariprasad, Sarika Esackimuthu, Saritha Madhavan, Rajalakshmi Sivanaiah, and Angel Deborah Suseelan. 2022. Ssn_mlrg1@dravidianlangtech-acl2022: Troll meme classification in tamil using transformer models. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 132–137, Dublin, Ireland. Association for Computational Linguistics.

Md Mithun Hossain, Md Shakil Hossain, M. F. Mridha, and Nilanjan Dey. 2026. A vision-language model for multitask classification of memes. *Neural Networks*, 194:108089.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. *arXiv preprint arXiv:2005.04790*.

Zhenkai Qin, Qining Luo, Zhidong Zang, and Hongpeng Fu. 2025. Detecting sarcasm in user-generated content integrating transformers and gated graph neural networks. *PeerJ Computer Science*, 11:e2817.

Saranya Rajiakodi, Shunmuga Priya Muthusamy Chinnan, Premjith B, Subalalitha CN, Rahul Ponnusamy, Anshid K A, Bhuvaneshwari Sivagnanam, Jananayagan V, Bharathi Raja Chakravarthi, Ragavan N, and Santhini P. 2026. Overview of the shared task on multilevel political meme classification in tamil and malayalam. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

CUET_InferX@DravidianLangTech 2026: Shared Task on Dialect Based Speech Recognition and Classification in Tamil

Md. Ashraful Islam Semon, Jihadul Islam, Ratnajit Dhar, Hasan Murad
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology, Bangladesh
{u2104059,u2104080,u2004008}@student.cuet.ac.bd, hasanmurad@cuet.ac.bd

Abstract

Tamil has a lot of internal variability, including the way it is used in casual conversations, code mixing, and phonetic differences in the way it is spoken in different regions, making it quite difficult to transcribe the spoken word and classify the dialects. In order to address these challenges, our paper presents the system developed by the CUET_InferX team for the Shared Task on Dialect Based Speech Recognition and Classification in Tamil, which was part of DravidianLangTech@ACL 2026. For Subtask 2 (ASR), our proposed system is based on a dual-architecture design that incorporates a fine-tuned Whisper-large-v3 model with Low-Rank Adaptation (LoRA) and a Wav2Vec2 XLSR-53 model, topped with a KenLM statistical language model for n-gram phonetic correction. Our ASR system resulted in a Word Error Rate (WER) of 0.54, which earned us 2nd position for Subtask 2. For Subtask 1 (Speech-Based Dialect Classification), our proposed system is based on a text-based weighted ensemble of IndicBERT, MuRIL, XLM-RoBERTa, and TamilBERT models, which is completely dependent on our ASR system's transcription outputs. Our proposed system achieved a Macro F1 score of 0.22, which earned us 9th position for Subtask 1.

1 Introduction

As voice technology grows, Automatic Speech Recognition (ASR) has also gained prominence as a key area of research. Despite recent progress in automatic speech recognition for high-resource languages, Tamil speech recognition is still difficult. This is due to dialect differences, complex structure, code-mixing, and a lack of balanced speech data, especially for spontaneous conversation.

The "Shared Task on Dialect Based Speech Recognition and Classification in Tamil: DravidianLangTech@ACL 2026" (Bharathi et al., 2026) is designed with the aim of creating a multi-dialect speech dataset (Bharathi et al., 2025) that challenges speech recognition models to perform well despite high intra-language variations. The objective is two-fold: speech classification into one of the four dialects such as Northern, Southern, Western, and Central, and accurate speech recognition into dialect-specific Tamil text.

In our case, we employed a dual-architecture acoustic system for ASR and a text-driven ensemble system for dialect classification. For ASR, we employed a fine-tuned Whisper-large-v3 model (Radford et al., 2023) using Low Rank Adaptation (LoRA) (Hu et al., 2021), along with a Wav2Vec2 XLSR-53 model (Conneau et al., 2020a), along with a statistical language model based on KenLM (Heafield, 2011). This system performed well, yielding a Word Error Rate of 0.54. For dialect classification, we employed a weighted ensemble of four text-based models IndicBERT, MuRIL, XLM-RoBERTa, and TamilBERT over the ASR-transcribed text. This system obtained a Macro F1 score of 0.22. The core contributions of our research work are as follows -

- We designed a dual architecture ASR pipeline that combines Whisper (with LoRA) and Wav2Vec2 XLSR-53 to robustly handle heavy dialect variations and code-mixed speech.
- We used a KenLM based n-gram rescoring method to maintain unnormalized dialect-specific colloquial spelling forms.
- A cascaded text-based ensemble was considered for the classification of dialects demonstrating the importance of acoustic features in the classification of dialects, in addition to textual features.

Detailed implementation information is available in the GitHub repository- https://github.com/semon87/shared-task-dialect-cuetinferx

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 201–206
July 3, 2026 ©2026 Association for Computational Linguistics

2 Related Work

The domain of Tamil ASR is shifting towards end-to-end deep learning with cross-lingual transfer learning. The DravidianLangTech shared task (Bharathi et al., 2026), built upon the multi-dialect Tamil speech corpus introduced by (Bharathi et al., 2025), provides an important benchmark for evaluating ASR systems under strong intra-language variability and spontaneous speech conditions.

Self-supervised acoustic models have shown strong effectiveness for Tamil ASR. (Akhilesh et al., 2022) fine-tuned XLSR-Wav2Vec2.0 using a CTC objective and reported a WER of 0.58 on Common Voice Tamil. Similarly, (Srinivasan et al., 2022) demonstrated that fine-tuned XLSR models obtain competitive performance in shared task settings. Focusing on Whisper-based architectures, (Saranya et al., 2024) observed substantial performance degradation of pre-trained Whisper models on unconstrained dialectal speech, but reduced the WER to 61% using LoRA-based adaptation. (Acharya et al., 2025) further validated the effectiveness of parameter-efficient fine-tuning of Whisper models in shared task scenarios, reinforcing the importance of adaptation for handling speaker and dialect variability.

Beyond acoustic modeling, the role of external language models in low-resource ASR has gained renewed attention. (Liu et al., 2024) showed that incorporating n-gram language models during decoding consistently reduces word error rates in low-resource languages.

For dialect classification, acoustic representations have been shown to capture critical phonetic and prosodic cues (Saranya et al., 2024). Text-based dialect classification relies on vocabulary and context patterns. The effectiveness of multilingual transformers like XLM-RoBERTa (Conneau et al., 2020b), IndicBERT (Kakwani et al., 2020), MuRIL (Khanuja et al., 2021) and TamilBERT (Joshi, 2022) has been demonstrated for NLP in Indian languages, which have a rich morphology and code-mixing capabilities.

3 Dataset

The data for the shared task (Bharathi et al., 2025) consists of a set of speech recordings in Tamil with their corresponding ASR transcripts and dialect information. Table 1 showing the training and test data, four dialects are present: Southern, Northern, Western, and Central.

Dialect Region	Training	Test
Southern	1427	212
Northern	1696	189
Western	1126	116
Central	885	62
Total	5134	579

Table 1: Distribution of training and test samples across dialect categories.

4 Methodology

4.1 Problem Formulation

The problem formulation for Subtask 2 (ASR) entails speech transcription with respect to mapping dialectical Tamil speech into transcripts considering colloquialisms. On the other hand, the formulation of Subtask 1 for dialect classification is done by means of cascade text classification using transcripts generated from ASR for predicting the four regional dialects, i.e., Northern, Southern, Western, and Central.

4.2 Data Preprocessing

For the acoustic models, we paired speech data with their respective transcripts and dialect regions, and the audio was converted to 16 kHz mono to match the pre-trained ASR model requirements. For mitigating the dialect imbalance, we have also tried basic audio augmentation techniques like time stretching and concatenating short utterances of minority dialect classes. Although the WER slightly decreases, it was excluded from the final pipeline due to increased computational overhead and GPU memory limitations.

For the classification of the dialect, the transcripts were tokenized based on the tokenizer for the respective model. The transcripts were then padded or truncated to a maximum of 128 tokens, and the dialect labels were converted into numerical form for training.

4.3 Experimented Models

4.3.1 Automatic Speech Recognition

To handle the dialectal variations of Tamil language, we implemented a dual-architecture ASR framework. We fine-tuned *whisper-large-v3*(with LoRA) optimized by Unsloth (Daniel Han and team, 2023) library on the provided corpus of Tamil dialects. LoRA adapters were applied to the attention projection layers (`q_proj` and `v_proj`) with rank $r = 64$, $\alpha = 64$, dropout $= 0$, and no bias pa-

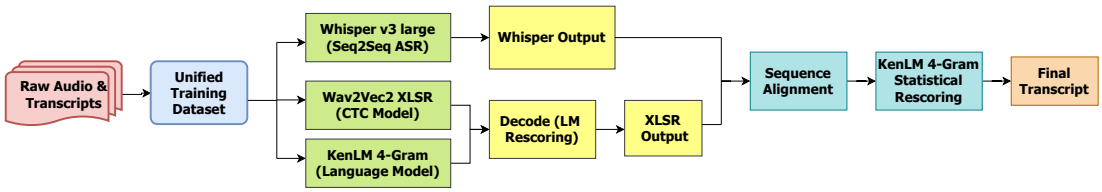


Figure 1: Proposed dual-architecture dialectal ASR pipeline

rameters, enabling computationally efficient training under limited GPU memory constraints. Whisper decoding was performed using beam search to obtain high quality transcripts. We fine-tuned Facebook's cross-lingual *wav2vec2-large-xlsr-53* model using Connectionist Temporal Classification (CTC). This model is specifically suited to handle the dialects and the phonetic variations in the speech patterns of the speakers. This model has also been fine-tuned on the provided dataset to suit the dialects and filtered too short (less than 1 sec) or too long (greater than 25 sec) audio. To further improve the transcription results, we trained a 4-gram statistical language model using KenLM on the training transcripts and used it to decode the transcripts. The KenLM decoding parameters were selected through empirical grid search on a held-out validation subset by evaluating various language-model weights (α) and word insertion bonuses (β) using WER as the selection criterion. A larger beam width was used during final decoding to improve search coverage, while smaller beams were used during parameter tuning for computational efficiency. Figure 1 shows the system architecture of ASR task.

4.3.2 Dialect Classification

In the case of dialect classification, we used a text-based approach by utilizing ASR transcripts. We have employed four different transformer-based architectures IndicBERT, MuRIL, XLM-RoBERTa, TamilBERT and trained with ASR transcripts of training data. After that, we used generated ASR transcript from Subtask 2 to classify the dialect region. Figure 2 shows the system architecture of dialect classification task.

4.4 Ensembling and Post-Processing

In ASR task, Whisper output and rescored XLSR model were combined with an edit-distance-style sequence alignment based on the Python SequenceMatcher. Each hypothesis was tokenized into Tamil words based on whitespace, without further morphological segmentation. Simply matching word

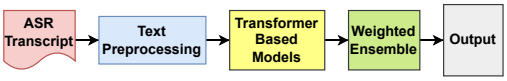


Figure 2: Proposed text-based dialect classification pipeline

Component	Key Settings
Whisper (Seq2Seq)	5 epochs, LR = 1e-4, Num of Beam = 5
XLSR (CTC)	30 epochs, Beam width = 500, α (LM weight) = 0.4, β (Word bonus) = 1.5
Language Model	KenLM 4-gram

Table 2: Hyperparameters for ASR and decoding

segments were directly copied to the combined final alignment; and conflicts due to substituted, inserted, or deleted segments were resolved with 4-gram KenLM rescoring. For each conflicting segment, normalized language model scores were computed and the segment with the higher contextual probability was used in the final alignment. A small bias toward XLSR outputs was additionally applied during conflict resolution, as XLSR showed better validation performance on dialect-specific phonetic variants.

For the classification of dialects, a four-model transformer ensemble is employed. In this approach, each model is used for the analysis of the transcript obtained from the ASR system. The contribution of each model is weighted according to its validation accuracy, and the final prediction is obtained via weighted soft voting.

4.5 Parameter Setting

Table 2 shows the best performing parameter settings for the ASR task. For dialect classification, all transformer models were trained for 15 epochs. In the weighted ensemble, XLM-RoBERTa got the highest weight of 1.4, L3Tamil-BERT followed with 1.1, IndicBERT and MuRIL both received weights of 0.9 based on their validation performance.

5 Results and Analysis

For dialect classification, the macro-averaged F1 score was used to evaluate the system's performance across the four dialect classes, while ASR performance was evaluated using the Word Error Rate (WER).

5.1 Comparative Analysis

Model	WER	CER
XLSR	0.6127	0.1779
Whisper	0.5510	0.1593
XLSR + KenLM	0.5491	0.1694
Whisper + Audio Augmentation	0.5434	0.1577
Whisper + XLSR + KenLM (Ensemble)	**0.5410**	0.1690

Table 3: Performance of different ASR systems on the test dataset

For ASR task, Table 3 shows that Whisper-large-v3 achieved a WER of 0.5510, while KenLM rescoring improved the XLSR model to 0.5491. Audio augmentation further improved Whisper performance to 0.5434. The best result came from using the Whisper + XLSR + KenLM ensemble, which achieved a WER of 0.5410.

For dialect classification task, Table 4 results suggest that the models tested on the Whisper ASR source outperform the ones tested on the ensemble ASR source. The best-performing model was the weighted ensemble model (XLM-RoBERTa, IndicBERT, MuRIL, TamilBERT) on the Whisper ASR source, with a macro F1 of 0.2209 and Figure 3 shows the confusion matrix.

5.2 Error Analysis

During sequence alignment, long compound words were often split inconsistently between the Whisper and XLSR models. This inconsistency led to localized insertion and deletion penalties in the final ensemble. In addition, the trade-off between preserving dialects within the transcription and improving overall transcription accuracy through KenLM rescoring was noted. Although KenLM rescoring reduced the overall Word Error Rate (WER) by focusing on contextually likely n-grams, it caused a normalization effect by often replacing rare dialect markers with standard vocabulary. As a result, regional linguistic features in the combined text were weakened. In contrast, the standalone Whisper outputs maintained these casual patterns more effectively. This explains why Whisper performed better in the downstream dialect classification task,

ASR Source	Model	F1
Whisper+XLSR +KenLM	IndicBERT	0.1635
	XLM-RoBERTa	0.0698
	Ensemble	0.1633
	Whisper + XLM-RoBERTa (Fusion)*	0.3839
Whisper	IndicBERT	0.1698
	XLM-RoBERTa	0.1153
	Ensemble	**0.2209**
	Whisper + XLM-RoBERTa (Fusion)*	0.4044

*Post-competition result.

Table 4: Performance of different dialect classification systems on the test dataset

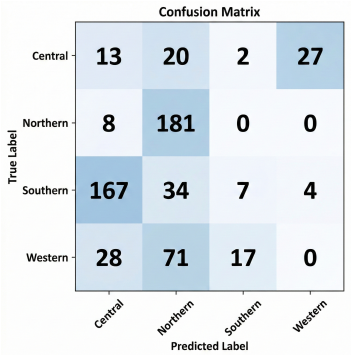


Figure 3: Confusion matrix of weighted ensemble process

even though it had a higher baseline ASR error rate.

Our primary submission focused on text-based dialect classification. This approach limited the model's ability to capture pronunciation and prosodic cues. In a post-competition experiment, a multimodal early-fusion model combined Whisper acoustic embeddings with XLM-RoBERTa text embeddings which achieved a macro F1 score of 0.40. This result shows how crucial acoustic features are for Tamil dialect classification.

6 Conclusion

In this work, we presented systems for dialect-based Tamil ASR and classification for DravidianLangTech@ACL 2026. Our Whisper XLSR KenLM ensemble achieved the best ASR performance with a WER of 0.5410. The weighted text-based dialect classification ensemble reached a macro F1 score of 0.2209. Experimental results show that transcript-only methods are not enough for strong dialect classification. Future work will focus on combining acoustic embeddings and increasing strength against ASR-induced errors.

Limitations

The ASR system can be improved by utilizing a larger and more varied speech corpus of multiple dialects and by employing sophisticated data augmentation techniques to improve its resilience against environmental noises and speech variations. For the dialect classification process, relying on the ASR transcript resulted in propagating errors and losing critical acoustic information such as intonation, prosody, and speech rates that are critical for dialect differentiation. Future studies should investigate multimodal models that combine acoustic and textual information to better capture dialect variations.

Ethical Considerations

In this study, we have developed our methodology following the highest ethical practices. By advancing dialect-aware speech recognition and classification in the Tamil language, we aim to develop speech technology that is more diverse and representative of the various dialects spoken in the regions. While there are chances of dialect identification tools being misused to profile or reinforce regional prejudices, we are committed to developing this technology for research and development purposes only. We are committed to developing artificial intelligence technology that is fair, transparent, and respectful of cultures, and to the ethical use of dialect-sensitive speech technology.

Acknowledgments

We extend our heartfelt thanks to the organizers of the Shared Task on Dialect Based Speech Recognition and Classification in Tamil at DravidianLangTech@ACL 2026 (Bharathi et al., 2026) for providing the multi-dialect corpus (Bharathi et al., 2025) and evaluation system. We also thank Kaggle for providing us with the computational resources to train our models.

References

Priyobroto Acharya, Soham Chaudhuri, Sayan Das, Dipanjan Saha, and Dipankar Das. 2025. Junlp@lt-edi-2025: Efficient low-rank adaptation of whisper for inclusive tamil speech recognition targeting vulnerable populations. In *Proceedings of the 5th Conference on Language, Data and Knowledge: Fifth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 17–25, Naples, Italy. Unior Press.

A Akhilesh, Brinda P, Keerthana S, Deepa Gupta, and Susmitha Vekkot. 2022. Tamil speech recognition using xlsr wav2vec2.0 & ctc algorithm. In *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–6. IEEE.

B. Bharathi, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, S. Saranya, and S. Suhasini. 2026. Findings in Tamil Dialect Speech Recognition and Classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

B Bharathi, S Saranya, P Vijayalakshmi, and T Nagarajan. 2025. Multi-dialect speech corpus creation for enhancing tamil automatic speech recognition. *Circuits, Systems, and Signal Processing*, pages 1–19.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2020a. Unsupervised cross-lingual representation learning for speech recognition. *arXiv preprint arXiv:2006.13979*.

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 ACL*.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

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 and 1 others. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Zoey Liu, Nitin Venkateswaran, Eric Le Ferrand, and Emily Prud'hommeaux. 2024. How important is a language model for low-resource asr? In *Findings of*

the Association for Computational Linguistics: ACL 2024, pages 206–213, Bangkok, Thailand. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28448–28481. PMLR.

S. Saranya, B. Bharathi, S. Gomathy Dhanya, and Aishwarya Krishnakumar. 2024. Real-time continuous tamil dialect speech recognition and summarization. *Circuits, Systems, and Signal Processing*, 44(4):2855–2881.

Dhanya Srinivasan, B. Bharathi, Thenmozhi Durairaj, and B. Senthil Kumar. 2022. Ssncse_nlp@lt-edi-acl2022: Speech recognition for vulnerable individuals in tamil using pre-trained xlsr models. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 317–320, Dublin, Ireland. Association for Computational Linguistics.

Cuet_Neural_Navigators@DravidianLangTech 2026: Depression Detection from Malayalam and Tamil Speech using Self-Supervised Acoustic Models

Shuva Dey, Abir Dey, Sha Newaz Mahmud, Hasan Murad
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology, Bangladesh
{u2104001,u2104005,u2004081}@student.cuet.ac.bd,
hasanmurad@cuet.ac.bd

Abstract

Depression detection from speech aims to find signs of depression using behavioral signals. This approach enables early mental health screening and makes it scalable. However, the task is tough because of subtle acoustic cues, differences among speakers, and language-specific patterns. In this work, we introduce our system for the Shared Task on Depression Detection in Dravidian Languages (DD-DL) at DravidianLangTech@ACL 2026. We focus on speech in Tamil and Malayalam. We explore pretrained self-supervised speech encoders, including HuBERT, XLS-R, and Whisper, to identify acoustic patterns related to depression directly from raw audio. Our method combines these models through ensembling to capture different acoustic features. The experiments use stratified evaluation and cross-lingual analysis to check how well the models work across languages. Results show that pretrained acoustic representations effectively capture vocal features of depression, achieving Macro-F1 scores of 0.9058 for Tamil and 0.9396 for Malayalam. However, cross-lingual transfer faces challenges because of phonetic and prosodic differences.

1 Introduction

Depression constitutes a significant worldwide mental health problem which requires immediate detection to enable appropriate treatment and assistance. While textual signals have been widely studied for automatic depression detection, speech offers additional insights through acoustic and paralinguistic elements including prosody, pitch, and speaking rate that reflect emotional states.

However, speech-based depression detection has not received sufficient research attention for low-resource Dravidian languages such as Tamil and Malayalam. The Shared Task on Depression Detection from Malayalam and Tamil Speech Data at DravidianLangTech@ACL 2026 (Jyothish Lal et al., 2026) introduces a benchmark dataset and evaluation framework for this problem, where the goal is to automatically determine whether a speaker shows depressive symptoms from their speech recordings.

We present our system based on pretrained self-supervised speech encoders used as frozen feature extractors, with a lightweight classification head trained on Tamil and Malayalam speech data to identify depression-related acoustic patterns. We examine how different evaluation settings affect model performance and study how speaker differences and language variations create obstacles.

Our research contributions include the following:

- We develop a depression detection system for Tamil and Malayalam speech using pretrained self-supervised acoustic representations.
- We evaluate model robustness under stratified and speaker-disjoint evaluation settings.
- We analyze cross-lingual transferability of depression-related acoustic representations across Dravidian languages.

For implementation details and access to the complete codebase, please refer to our GitHub repository. [1]

2 Related Work

Automatic depression detection has been explored using textual, acoustic, and multimodal signals. Surveys highlight the growing role of AI in identifying depression-related patterns while noting challenges such as annotation noise and cultural variability (Babu and Kanaga, 2021). Transformer-based models have been applied for depression detection in Tamil text (Hemalatha et al., 2025).

[1] https://github.com/Snikdrek/Deppression-Detection-of-Dravidian-speech-Tamil-Malayalam-CUET_Neural_Networks

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 207–211
July 3, 2026 ©2026 Association for Computational Linguistics

Speech-based approaches utilize paralinguistic cues such as prosody and pitch variation to capture depressive characteristics. Recent work shows that acoustic features alone can distinguish depressive speech across languages (Binu et al., 2024). The InStant-EMDB corpus introduced English–Malayalam speech data for depression detection (Mathew et al., 2024), while other studies explored multimodal systems (Reddy et al., 2024) and feature-fusion methods for Dravidian speech classification (Kritika et al., 2025).

Shared tasks have further advanced low-resource Dravidian language research (Premjith et al., 2024; Anilkumar et al., 2026), with earlier text-based depression detection tasks demonstrating the effectiveness of transformers (Sampath et al., 2023). The DravidianLangTech@ACL 2026 shared task (Jyothish Lal et al., 2026) provides a standardized speech-based benchmark. Our work builds on these efforts using pretrained self-supervised speech encoders.

3 Dataset

We use the dataset released for the Shared Task on Depression Detection from Malayalam and Tamil Speech Data at DravidianLangTech@ACL 2026 (Jyothish Lal et al., 2026). Depressed utterances were recorded at 16 kHz (2–5 seconds each), while non-depressed recordings were captured at 48 kHz and resampled for consistency. Tables 1 and 2 summarize the data distribution and speaker statistics.

Language	Train	Test	Total
Tamil	1374	160	1534
Malayalam	1888	200	2088

Table 1: Train–test distribution.

Parameter	Malayalam	Tamil
Total Samples	1,888	1,534
Depressed Samples	888	534
Unique Speakers	8	9
Depressed Speakers	3	4
Non-depressed Speakers	5	5

Table 2: Class and speaker statistics across languages.

4 Methodology

Our approach models depression detection as a binary classification task over raw speech signals. Given an input utterance x, the objective is to predict a label $y \in \{0, 1\}$, where 1 denotes depressed and 0 denotes non-depressed speech. Figure 1 illustrates the overall architecture of the proposed framework.

4.1 Preprocessing

All audio recordings were resampled to 16 kHz for consistency. Depressed samples were originally recorded at 16 kHz while non-depressed samples were recorded at 48 kHz; resampling ensured uniform input representation. Each utterance was truncated or padded to 5 seconds to standardize input length. No handcrafted acoustic features were used; models operate directly on raw waveforms.

4.2 Speech Representation Models

We employ pretrained self-supervised speech encoders to obtain acoustic representations:

- **HuBERT** (Hsu et al., 2021): A self-supervised speech representation model trained via masked prediction of discrete speech units.[2]
- **XLS-R** (Babu et al., 2022): A multilingual extension of wav2vec 2.0 pretrained on large-scale cross-lingual speech data.[3]
- **Whisper** (Radford et al., 2023): A transformer-based multilingual speech model trained on large-scale audio data.[4]

Pretrained backbones were initialized with publicly available weights. To reduce overfitting on the small dataset, encoder parameters were frozen and only a lightweight classification head was trained.

4.3 Classification Layer

Hidden representations from each encoder were mean-pooled across the temporal dimension and passed to a linear classification layer. Cross-entropy loss was used for optimization with the AdamW optimizer. Gradient accumulation was applied to simulate larger batch sizes.

4.4 Ensemble Strategy

To improve robustness, predictions from HuBERT, XLS-R, and Whisper were combined via logit-level averaging:

[2] https://huggingface.co/facebook/hubert-base-ls960

[3] https://huggingface.co/facebook/wav2vec2-xls-r-300m

[4] https://huggingface.co/openai/whisper-base

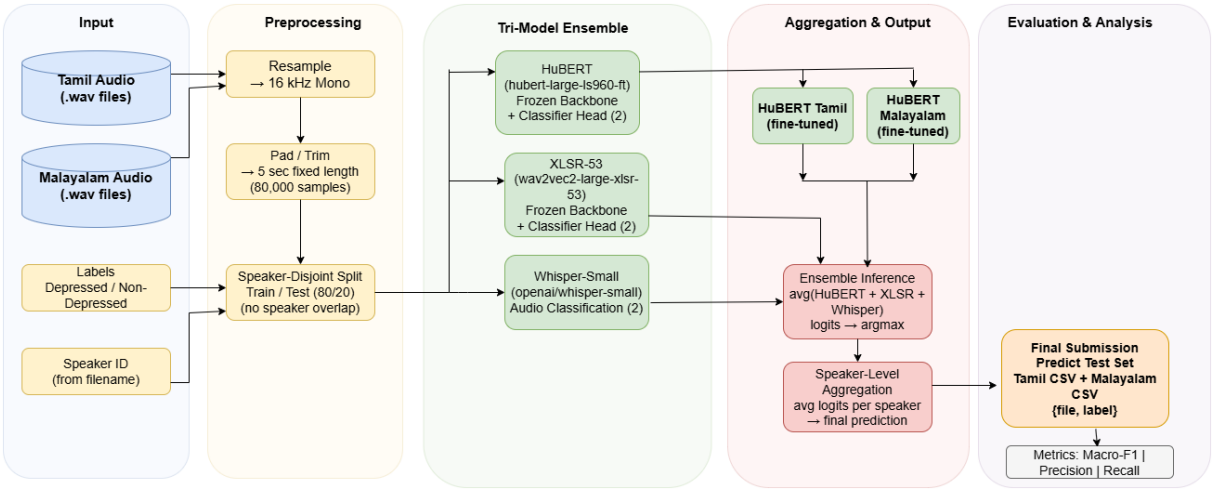


Figure 1: Overview of the proposed depression detection framework.

$$\hat{y} = \frac{1}{3}(z_{\text{HuBERT}} + z_{\text{XLS-R}} + z_{\text{Whisper}})$$

where z denotes model logits. Final predictions were obtained by applying the softmax function to the averaged logits.

4.5 Evaluation Protocol

Experiments were conducted under two evaluation settings:

- **Stratified Split**: Standard 80/20 train–test split preserving label distribution. No fixed random seed was used during splitting, which may cause minor variations across runs.
- **Speaker-Disjoint Split**: Train and test sets were constructed with non-overlapping speakers to evaluate speaker generalization.

Cross-lingual experiments were also performed by training on one language and evaluating on the other. Macro-F1 was used as the primary evaluation metric due to class imbalance.

4.6 Speaker-Level Aggregation

Since depression is a speaker-level condition, utterance-level predictions were aggregated by averaging logits across all utterances from the same speaker before computing final predictions.

4.7 Implementation Details

Models were implemented using `PyTorch` and the `HuggingFace Transformers` library. Training used the AdamW optimizer with a learning rate of 1×10^{-4} for 20 epochs and a per-device batch size of 1. Gradient accumulation with 8 steps produced an effective batch size of 8. Mixed-precision (FP16) training was enabled when supported. Model checkpoints were evaluated each epoch, and early stopping based on validation Macro-F1 retained the best checkpoint. Encoder backbones were frozen and only the classification head was updated.

5 Results and Analysis

In this section, we evaluate our models under monolingual stratified, speaker-disjoint, and cross-lingual settings. Performance is measured using Accuracy, Precision, Recall, and Macro-F1, with Macro-F1 used as the primary metric due to potential class imbalance.

The official shared-task submission achieved Macro-F1 scores of 0.8542 for Tamil and 0.9345 for Malayalam on the test set, ranking **4th** on the shared task leaderboard.

All results reported in the following tables correspond to our experimental runs, which may show slight variations due to stochastic training and data splitting.

5.1 Model Comparison

Table 3 presents the performance of the HuBERT baseline and the proposed ensemble model for Tamil and Malayalam under stratified evaluation. The ensemble model consistently outperforms the single-model baseline in both languages.

For Tamil, the ensemble achieves a Macro-F1 score of 0.9058, improving over the HuBERT baseline by approximately 1.3%. Similarly, for Malayalam, the ensemble obtains the best performance

Experiment	Accuracy	Macro-F1	Precision	Recall
Tamil HuBERT (Stratified GT)	0.8938	0.8930	0.9044	0.8938
Tamil Ensemble (Stratified GT)	**0.9063**	**0.9058**	**0.9141**	**0.9063**
Malayalam HuBERT (Stratified GT)	0.9250	0.9243	0.9359	0.9235
Malayalam Ensemble (Stratified GT)	**0.9400**	**0.9396**	**0.9474**	**0.9388**
Tamil (Speaker-Disjoint)	–	0.8288	0.8202	0.8839
Malayalam (Speaker-Disjoint)	–	0.8157	0.7848	0.9137
Tamil → Malayalam (Cross-lingual)	0.6150	0.6142	0.6149	0.6143
Malayalam → Tamil (Cross-lingual)	0.7688	0.7557	0.8419	0.7688

Table 3: Performance comparison across monolingual, speaker-disjoint, and cross-lingual evaluation settings.

with a Macro-F1 score of 0.9396. These results indicate that combining multiple acoustic models helps capture complementary speech representations, improving depression detection performance.

5.2 Generalization Analysis

To assess robustness across unseen speakers, we conduct speaker-disjoint experiments where training and testing speakers do not overlap. Performance drops compared to stratified evaluation, with Macro-F1 scores of 0.8288 for Tamil and 0.8157 for Malayalam, highlighting challenges in generalizing across speakers. Nevertheless, recall remains relatively high, especially for Malayalam (0.9137), indicating strong sensitivity to depressed speech.

We also evaluate cross-lingual transfer between Tamil and Malayalam. Training on Malayalam and testing on Tamil achieves a Macro-F1 of 0.7557, while the reverse direction achieves 0.6142. This asymmetry suggests that depression-related acoustic patterns learned from Malayalam transfer more effectively to Tamil.

5.3 Error Analysis

Figures 2 and 3 present the confusion matrices for the stratified evaluation setting. For Malayalam, the model correctly classifies 102 non-depressed and 83 depressed samples, while for Tamil it correctly predicts 78 non-depressed and 67 depressed samples, indicating strong class separation.

Most misclassifications occur near the boundary between depressed and non-depressed speech, likely due to subtle acoustic similarities between certain utterances.

6 Conclusion

We presented a speech-based depression detection system for Tamil and Malayalam in the DravidianLangTech@ACL 2026 Shared Task. Using pretrained self-supervised speech encoders (HuBERT, XLS-R, and Whisper), our approach learns

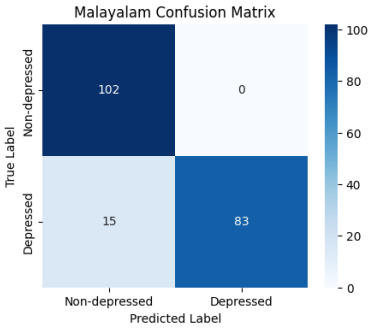


Figure 2: Confusion matrix for Malayalam under stratified evaluation.

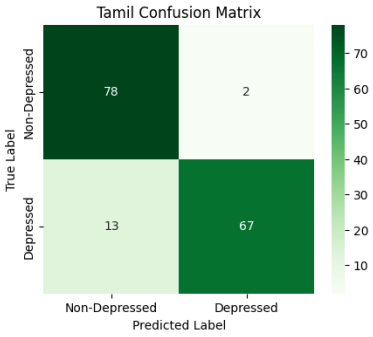


Figure 3: Confusion matrix for Tamil under stratified evaluation.

depression-related acoustic patterns directly from raw speech.

The ensemble model achieved strong monolingual performance, reaching Macro-F1 scores of 0.9058 for Tamil and 0.9396 for Malayalam. The evaluation results show that both speaker-disjoint and cross-lingual tests produced lower scores, which demonstrate the system's difficulties with speaker generalization and its ability to handle different languages.

The study demonstrates that multilingual speech models pre-trained on multiple languages work effectively for depression detection in low-resource environments while showing that existing systems need better performance with various speakers and languages.

Limitations

This work is limited by the relatively small dataset and the small number of unique speakers, which affects generalization, particularly in speaker-disjoint settings. The collected recordings used short utterances which were produced in controlled environments and do not represent the full range of real-world conversational variations.

The training process used frozen pretrained encoders to prevent overfitting, which created a problem for adapting to specific language acoustic features. The performance gaps between Tamil and Malayalam indicate existing transferability problems between the two languages.

Ethics Statement

Depression detection from speech involves sensitive mental health data. The proposed system is not designed for clinical diagnosis but functions as a professional supervision screening tool.

The actual deployment requires organizations to evaluate three main factors including the risks of misclassifying data and the need to protect user privacy and the possibility of demographic biasing. Responsible use requires informed consent from participants, clear communication of the system's limitations, and qualified professional oversight at every stage of deployment.

References

A. Anilkumar, G. Jyothish Lal, B. Premjith, and B. R. Chakravarthi. 2026. Dravlangguard: A multimodal approach for hate speech detection in dravidian social media. In *Speech and Language Technologies for Low-Resource Languages*, volume 2656 of *Communications in Computer and Information Science*, Cham. Springer.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Gober, Romi Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2022. Xls-r: Self-supervised cross-lingual speech representation learning at scale. In *Proceedings of Interspeech*.

Nirmal Varghese Babu and E Grace Mary Kanaga. 2021. Sentiment analysis in social media data for depression detection using artificial intelligence: A review. *Journal of Healthcare Engineering*, 2021.

Sona Binu, Jismi Jose, Fathima Shimna K V, and 1 others. 2024. Language-agnostic analysis of speech depression detection. *arXiv preprint arXiv:2409.14769*.

M. Hemalatha, R. Varshini, Devananda Anil, and E. Janani. 2025. Towards inclusive AI: Deep learning based depression detection with transformers RNN-LSTM. *Journal of Emerging Technologies and Innovative Research*.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Benber, and Mohamed Abdelrahman. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

G. Jyothish Lal, B. Premjith, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Thenmozhi Durairaj, and Prasanna Kumar Kumaresan. 2026. Shared task on depression detection from malayalam and tamil speech data. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

A. Kritika, S. Meenakshy, Arya Palackal Shijish, Riya Rajeev, and G. Jyothish Lal. 2025. Dravimood: Speech-based depression classification in dravidian languages using feature fusion and deep learning. In *Proceedings of the Fourth International Conference on Speech and Language Technologies for Low-Resource Languages (SPELLL 2025)*.

Anjali Mathew, Raniya, Harsha Sanjan, Amjith S B, and 1 others. 2024. Instant-emdb: A multi model spontaneous english and malayalam speech corpora for depression detection. In *Proceedings of the IEEE Conference*.

B. Premjith, G. Jyothish, V. Sowmya, and B. Bharathi. 2024. Findings of the shared task on multimodal social media data analysis in dravidian languages (msmda-dl)@dravidianlangtech 2024. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 56–61.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, pages 28492–28518.

B Surekha Reddy, Jishitha Kondaveti, V Akshaya Bhavani, and P Aishwarya. 2024. Development of a depression detection system using speech and text data. EasyChair Preprint.

Kayalvizhi Sampath, Durairaj Thenmozhi, Bharathi Raja Chakravarthi, and 1 others. 2023. Overview of the shared task on detecting signs of depression from social media text. In *Proceedings of the Workshop on Language Technology for Equality, Diversity and Inclusion*. Association for Computational Linguistics.

CUET_SYNTHETICA@DravidianLangTech 2026: Multi Architecture Transformer Ensemble for Detecting Abusive Tamil Text Targeting Women

Miftahul Jannat Rishta, Sumaiya Zaman, Shiti Chowdhury, Hasan Murad
Department of Computer Science and Engineering,
Chittagong University of Engineering and Technology, Bangladesh
{u2104019, u2104110, u2004027}@student.cuet.ac.bd, hasanmurad@cuet.ac.bd

Abstract

Abusive language targeting women has been a serious problem on Tamil social media and building systems to detect it automatically is harder than it looks. Tamil is morphologically complex, people have written it mixed with English in ways no dictionary has accounted for and a lot of the hostility has been indirect enough that has slipped past models trained on surface patterns. In the Shared Task on Abusive Tamil Text Targeting Women on Social Media DravidianLangTech@ACL 2026, we have worked on classifying Tamil YouTube comments as Abusive or Non-Abusive. We have trained three transformer models four times each with different learning rates, giving us 12 models total. Their predicted probabilities have been averaged to make the final decision. The 12-model ensemble has achieved a macro F1 of 0.8086, outperforming all individual models and securing 4th place in the shared task. Combining Tamil-specialized and multilingual transformer models has outperformed any single-architecture approach.

1 Introduction

Social media has made it far easier for people to connect but also far easier to harass them and women have borne more than their share of that harassment (Sivagnanam et al., 2026). Abusive comments have not just stay on screen rather they have pushed people out of conversations and have made online spaces feel hostile (Premjith et al., 2024). Detecting this content automatically is one concrete thing NLP can do about it.

Tamil has made that harder than it sounds. It is spoken by tens of millions yet has remained poorly covered in NLP research. The language is morphologically complex, users have written it heavily mixed with English and much of the abuse is culturally specific enough that surface-level models miss it entirely (Chakravarthi et al., 2022). Annotated Tamil datasets, which are also small in relation to the scale of the problem (Sreelakshmi et al., 2024). Despite these challenges, recent advancements in multilingual models and cross-lingual embeddings have shown promise in improving Tamil NLP but much work remains to be done to fully capture its complexities.

The DravidianLangTech@ACL 2026 shared task (Sivagnanam et al., 2026) has provided us a benchmark of Tamil YouTube comments to classify as Abusive or Non-Abusive. We have combined MuRIL (Khanuja et al., 2021) and IndicBERT v2 (Doddapaneni et al., 2023) for Tamil-specific coverage with XLM-RoBERTa-Large (Conneau et al., 2020) for a wider multilingual signal, having trained each architecture four times with different learning rates to get twelve models in total. Their predicted probabilities have been averaged for the final decision, achieving a macro F1 of 0.8086.

Our main contributions are:

- We have developed a heterogeneous ensemble of Tamil-specialized and multilingual transformers that has outperformed any single-model baseline.
- We have implemented learning-rate diversity across four runs per architecture to introduce prediction variance.

Code is available at: https://github.com/Mif-taa/Abusive-Tamil-Text.

2 Related Work

Most early work on abusive language detection has been done in English and the jump to Dravidian languages has been slow. Tamil social media text has proven genuinely difficult as users have mixed Tamil and English mid-sentence and have relied on cultural references that are hard to annotate systematically. Chakravarthi et al. (2022) has addressed this by releasing DravidianCodeMix, which has put Tamil, Malayalam and Kannada offensive language

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 212–216
July 3, 2026 ©2026 Association for Computational Linguistics

detection on a shared benchmark for the first time. Dowlagar and Mamidi (2021) and Yasaswini et al. (2021) have shown that transformer fine-tuning can work for these languages, though class imbalance and code-mixing have remained difficult.

Indic-specific pretraining has made a real difference. Khanuja et al. (2021) has built MuRIL on 17 Indian languages and their transliterations - which matters because Tamil speakers on social media have often written Tamil words in Roman script. Doddapaneni et al. (2023) has extended this with IndicBERT v2 across 24 languages on a much larger corpus. Conneau et al. (2020) has shown that XLM-RoBERTa (Liu et al., 2019), scaled to 100 languages has transferred reasonably even to languages it has seen little of at training time. Sreelakshmi et al. (2024) has compared several of these models on Dravidian hate speech data and the DravidianLangTech shared tasks (Premjith et al., 2024; Sivagnanam et al., 2026) have pushed the field forward with standardized benchmarks. In the 2025 edition, Rahman et al. (2025) has found that combining MuRIL with XLM-BERT has outperformed either alone. Hanif and Rahman (2025) has found that learning rate choice matters more than model size and Thavarasa et al. (2025) has shown that implicit abuse in YouTube comments has been the hardest to catch - findings that have directly shaped our design. However, Tamil annotated data has remained limited and detecting abuse that relies on tone rather than explicit words has still been difficult, challenges we have faced in this work as well.

3 Data Description

The dataset has been sourced from the shared task on Abusive Tamil Text Targeting Women on Social Media-DravidianLangTech@ACL 2026. It has consisted of Tamil YouTube comments annotated with binary labels: Abusive and Non-Abusive. The training set has contained 3,652 samples and the test set 913. Some labels have had capitalization inconsistencies which we have corrected before training. The data distribution has shown in Table 1.

Label	Train	Test
Abusive	1,769	441
Non-Abusive	1,883	472
Total	**3,652**	**913**

Table 1: Dataset distribution after label normalization.

4 Methodology

4.1 Problem Formulation

Each Tamil YouTube comment has needed to be assigned one of two labels: Abusive or Non-Abusive. Rather than betting on a single model to make that call, we have trained $K = 12$ transformer classifiers and letting them vote. Each model i has produced a softmax probability $p_i(k \mid t)$ for comment t and class $k \in \{0, 1\}$. The final label is whichever class has received the highest average probability across all 12 models:

$$\hat{y} = \arg\max_k \frac{1}{K} \sum_{i=1}^{K} p_i(k \mid t) \quad (1)$$

This soft-voting strategy has kept individual model confidence in the picture rather than just counting votes, which has mattered when models have disagreed on borderline cases.

4.2 Preprocessing

The raw training data has contained 3,652 samples. After removing 401 duplicates, 3,251 clean samples have remained. An 80/20 stratified split has yielded 2,600 training and 651 validation samples. Deduplication has been applied to training data only. Each comment has been cleaned through an 11-step pipeline: HTML entity decoding, URL removal, email and mention removal, hashtag normalization, punctuation reduction, whitespace normalization, standalone number removal, quote normalization and zero-width character removal. Labels have been standardized to consistent casing. The class distribution has been near-balanced (51.3% Non-Abusive, 48.7% Abusive), so mild imbalance has been handled through class-weight adjustment rather than resampling.

4.3 Transformer Models

We have experimented with three transformer architectures. MuRIL has been pre-trained on 17 Indian languages and their transliterations, making it well-suited for Tamil social media text where users frequently romanize Tamil words. IndicBERT v2 has been trained on a large-scale Indic corpus covering 24 languages, providing strong monolingual Tamil representations. XLM-RoBERTa-Large has covered 100 languages and has contributed broad cross-lingual signal that complements the Indic-specialized models. Among individual runs, IndicBERT_2 has delivered the strongest single-model performance on the test set.

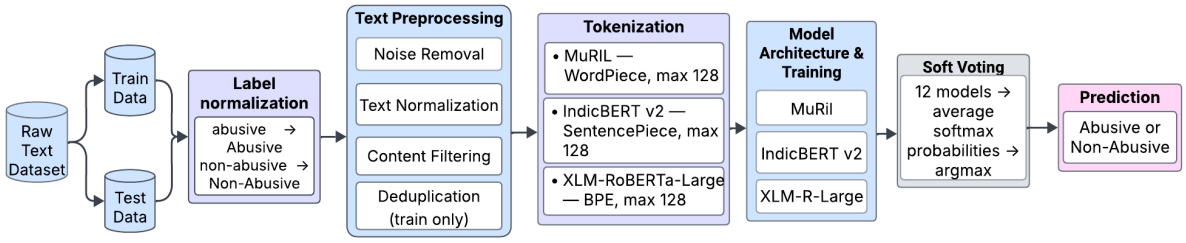


Figure 1: End-to-end pipeline of the heterogeneous 12-model ensemble for abusive Tamil text detection.

4.4 Ensemble

We have trained four runs per architecture by varying the learning rate, yielding 12 models total. Each model has been fine-tuned using an 80/20 train-validation split with class-weighted cross-entropy loss and the AdamW optimizer. Final predictions has been obtained by averaging the softmax probabilities across all 12 models. A soft-voting approach that accounts for prediction confidence rather than treating all votes equally. Figure 1 illustrates the overall process flow.

4.5 Evaluation Metrics

Macro F1, precision and recall have been used to evaluate all models, ensuring balanced performance across both the Abusive and Non-Abusive classes.

5 Results and Analysis

Table 2 has shown that among individual models, IndicBERT_2 has posted the strongest result F1: 0.8075, while the 12-model ensemble has outperformed all individual models with a macro F1 of 0.8086.

Model	Acc	P	R	F1
Best Single Models				
MuRIL_3	0.7963	0.7975	0.7949	0.7954
XLM-R-L_3	0.7919	0.7919	0.7913	0.7915
IndicBERT_2	0.8083	0.8096	0.8070	0.8075
Final Ensemble				
All 12 Models	**0.8094**	**0.8106**	**0.8082**	**0.8086**

Table 2: Best individual model per architecture and final 12-model ensemble on the official test set.

Table 3 has shown that no single-architecture ensemble has matched the full 12-model heterogeneous ensemble, confirming that architectural diversity has contributed beyond learning-rate variation alone. The final 12-model soft vote has pushed F1 to 0.8086, above any single model or single-architecture ensemble.

Architecture	Acc	P	R	F1
MuRIL (4 models)	0.7952	0.7949	0.7948	0.7948
XLM-R-L (4 models)	0.7996	0.7993	0.7991	0.7992
IndicBERT (4 models)	0.7985	0.8011	0.7964	0.7970
All 12 Models	**0.8094**	**0.8106**	**0.8082**	**0.8086**

Table 3: Ablation: per-architecture 4-model ensembles vs. final heterogeneous 12-model ensemble.

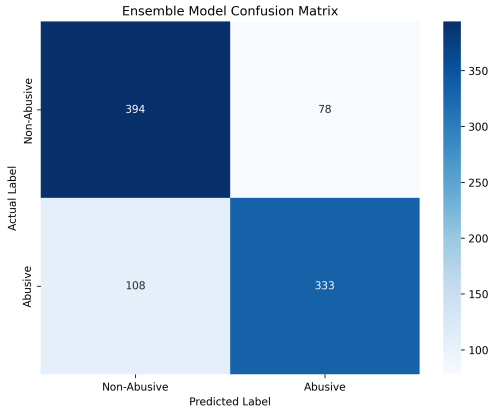


Figure 2: Confusion matrix for the final 12-model ensemble on the test set.

Figure 2 has shown the confusion matrix for the final ensemble. The model has correctly identified 394 non-abusive and 333 abusive comments, while producing 78 false positives and 108 false negatives. The higher false negative count suggests the ensemble has more often missed abusive content than over-flagged normal comments-a gap we would prioritize closing in future work.

5.1 Hyperparameter Settings

Table 4 has listed the training configuration for each model family. XLM-RoBERTa-Large has required much smaller learning rates (2e-6 to 5e-6) than MuRIL and IndicBERT v2. It is the largest model of the three and more sensitive to high learning rates. All models have used a cosine schedule with 10% warmup and AdamW with weight decay 0.01, training for 5/10 epochs.

Configurations	MuRIL	XLM-R-Large	IndicBERT v2
LR	4e-5	4e-6	3e-5
Batch	32	8	8
Epochs	10	10	5
Max Len	128	128	128
Optimizer	AdamW	AdamW	AdamW
W. Decay	0.01	0.01	0.01
Warmup	10%	10%	10%
Val Split	20%	20%	20%
Seeds	3407	3407	42

Table 4: Training configuration for 3 best model.

All experiments have been conducted on a Kaggle-provided NVIDIA Tesla T4 GPU (15.6 GB memory).

Architecture	R1	R2	R3	R4
MuRIL	2e-5	3e-5	4e-5	5e-5
XLM-R-Large	2e-6	3e-6	4e-6	5e-6
IndicBERT v2	2e-5	3e-5	2e-5	3e-5

Table 5: Learning rates across all 12 runs.

6 Conclusion

This work has addressed abusive Tamil text detection in the DravidianLangTech@ACL 2026 shared task, highlighting the challenges posed by morphological richness, code-mixing and culturally nuanced expressions. Our heterogeneous ensemble of MuRIL, XLM-RoBERTa-Large and IndicBERT v2 has outperformed all individual models on the official test set, demonstrating that combining Tamil-specialized and multilingual architectures has yielded more reliable predictions than relying on a single model. Although the performance gains have been consistent, they have been modest, indicating that learning-rate variation alone has provided limited ensemble diversity. Future research should explore greater model diversity through different random seeds, advanced ensemble strategies such as stacking or weighted voting, larger Tamil-focused pretrained models and improved modeling of implicit or culturally contextual abuse.

Error Analysis

The ensemble has struggled most with indirect and sarcastic abuse. The confusion matrix has shown 108 false negatives against 78 false positives, meaning the model has been more likely to miss abuse than over-flag normal content. False negatives have fallen into three categories: implicit abuse (hostile intent carried through cultural context), sarcastic comments (irony masking abusive intent) and code-mixed abuse (hostility appearing in the English portion of mixed comments). Representative false negatives have included comments containing culturally derogatory terms misread as neutral and sarcastic comments where tone has carried the hostility rather than explicit words. Future work should address these through sarcasm-aware modeling and larger Tamil datasets.

Limitations

The training set has been small at 3,251 samples with no labeled development set, so ensemble weights have been equal rather than tuned. Training 12 transformers is computationally expensive and all three architectures have shared similar pretraining objectives which has limited diversity and has likely kept gains modest. Full hyperparameter optimization using tools such as Weights & Biases has not been pursued due to the computational cost of running sweeps across 12 transformer models within the shared task timeline.

Ethical Statement

This work has used publicly available data from the shared task organizers. No personal information has been collected or stored. The dataset has contained harmful content targeting women and we have handled it carefully. We are aware that misclassification carries real costs and our goal has been fairness across both classes rather than metric optimization.

Acknowledgement

We have been grateful to the DravidianLangTech@ACL 2026 organizers for having provided the dataset and evaluation framework. We have thanked the developers of MuRIL, IndicBERT v2 and XLM-RoBERTa-Large whose open-source models have made this work possible. We have also acknowledged Devlin et al., Liu et al. and Chakravarthi et al. for the foundational work this research has built on.

References

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John P. McCrae. 2022. Dravidiancodemix: Sentiment analysis and offensive language identification dataset for dravidian languages in code-mixed text. *Language Resources and Evaluation*, 56(3):765–806.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco

Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2023. Towards leaving no indic language behind: Building monolingual corpora, benchmark and models for indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 12402–12426. Association for Computational Linguistics.

Suman Dowlagar and Radhika Mamidi. 2021. OFFLangOne@DravidianLangTech-EACL2021: Transformers with the class balanced loss for offensive language identification in Dravidian code-mixed text. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 154–159, Kyiv. Association for Computational Linguistics.

Tareque Md Hanif and Md Rashadur Rahman. 2025. CUET_Agile@DravidianLangTech 2025: Fine-tuning transformers for detecting abusive text targeting women from Tamil and Malayalam texts. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

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. In *arXiv preprint arXiv:1907.11692*.

B Premjith, Bharathi Raja Chakravarthi, Prasanna Kumar Kumaresan, Saranya Rajiakodi, Sai Karnati, Sai Mangamuru, and Chandu Janakiram. 2024. Findings of the shared task on hate and offensive language detection in telugu codemixed text (hold-telugu)@ dravidianlangtech 2024. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 49–55.

Md Mizanur Rahman, Srijita Dhar, Md Mehedi Hasan, and Hasan Murad. 2025. MSM_CUET@DravidianLangTech 2025: XLM-BERT and MuRIL based transformer models for detection of abusive Tamil and Malayalam text targeting women on social media. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Bhuvaneswari Sivagnanam, Kathiravan Pannerselvam, Jananayagan V, Charmathi Rajkumar, Ramesh Kannan R, Ratnavel Rajalakshmi, Shunmuga Priya Muthusamy Chinnan, Saranya Rajiakodi, and Bharathi Raja Chakravarthi. 2026. From Comments to Harm: A Findings Report on Abusive Tamil Text Targeting Women on Social Media. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

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.

Luxshan Thavarasa, Sivasuthan Sukumar, and Jubeerathan Thevakumar. 2025. Incepto@DravidianLangTech 2025: Detecting abusive Tamil and Malayalam text targeting women on YouTube. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Konthala Yasaswini, Karthik Puranik, Adeep Hande, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. 2021. IIITT@DravidianLangTech-EACL2021: Transfer learning for offensive language detection in Dravidian languages. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 187–194, Kyiv. Association for Computational Linguistics.

CUET_SYNTHETICA@DravidianLangTech 2026: Multilingual Transformer Based Hope Speech Detection for Coarse and Fine-Grained Classification in Tulu

Sumaiya Zaman, Miftahul Jannat Rishta, Shiti Chowdhury, Hasan Murad
Department of Computer Science and Engineering,
Chittagong University of Engineering and Technology, Bangladesh
{u2104110, u2104019, u2004027}@student.cuet.ac.bd, hasanmurad@cuet.ac.bd

Abstract

Hope speech has played a vital role in online communities, yet most NLP work has focused on English and a few high-resource languages, leaving code-mixed varieties like Tulu largely unexplored. In the Shared Task on Hope Speech Detection in Code-Mixed Tulu at DravidianLangTech@ACL 2026, we have tackled two subtasks: (i) coarse-grained classification into Encouraging, Discouraging, Uninvolved and Blended categories (Task 1) and (ii) fine-grained classification into Optimistic, Realistic, Inspiring, Fading and Hopelessness (Task 2). We have fine-tuned three multilingual transformer encoders XLM-RoBERTa-base, MuRIL and mBERT on the official training splits. In Task 1, a three-way soft-voting ensemble of all three models has yielded the best performance with a macro F1 of 0.58, securing 1st place. In Task 2, XLM-RoBERTa-base alone has outperformed both MuRIL and mBERT, achieving a macro F1 of 0.42 and also securing 1st place.

1 Introduction

Social media has made it easier to share support, but most NLP work has focused on detecting negative language rather than positive. Hope speech has received far less attention, particularly in low-resource code-mixed languages (Chakravarthi, 2020; Chakravarthi and Muralidaran, 2021). Tulu, a Dravidian language spoken in coastal Karnataka, has been especially underrepresented, with users writing it heavily mixed with English and Kannada on YouTube (Chakravarthi, 2020; Sundar et al., 2021; Chakravarthi et al., 2022).

The DravidianLangTech@ACL 2026 Shared Task has introduced the first gold-standard Tulu dataset for two subtasks (Thenmozhi et al., 2026; Chakravarthi and Muralidaran, 2021; Chakravarthi et al., 2022): coarse-grained classification into Encouraging, Discouraging, Uninvolved and Blended in Task 1 and fine-grained classification into Optimistic, Realistic, Inspiring, Fading and Hopelessness in Task 2. We have fine-tuned XLM-RoBERTa-base, MuRIL and mBERT, achieving macro F1 of 0.58 (Task 1) and 0.42 (Task 2), both securing 1st place. Our main contributions are:

- We have implemented an ensemble of pretrained models, including XLM-R, MuRIL and mBERT for coarse-grained hope tone classification, demonstrating the power of combining models for superior performance.
- We have utilized XLM-RoBERTa-base, a pretrained transformer model, for fine-grained classification of code-mixed Tulu text, showing the efficacy of fine-tuning a single model for detecting nuanced emotional expressions.

Detailed implementation information is available in the GitHub repository: https://github.com/sumaiyaa110/Hope-Speech-Detection

2 Related Work

Hope speech detection has evolved from monolingual datasets to multilingual resources through LT-EDI workshops (Chakravarthi, 2020; Chakravarthi and Muralidaran, 2021; Chakravarthi, 2022). Early work on English, Tamil and Malayalam has shown that cross-lingual transformers outperform traditional classifiers on code-mixed Dravidian text (Sundar et al., 2021; Malik et al., 2023). Subsequent systems have demonstrated that context-aware embeddings improve multilingual hope speech detection (Aggarwal et al., 2022; Das et al., 2023), while PolyHope has extended this to English tweets (Balouchzahi et al., 2023; Sidorov et al., 2025) and recent work has explored hierarchical attention and capsule-based models (Rehman et al., 2025). However, regional code-mixed languages like Tulu remain underexplored, a gap addressed by the DravidianLangTech@ACL 2026 Shared Task (Thenmozhi et al., 2026).

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 217–221
July 3, 2026 ©2026 Association for Computational Linguistics

3 Data Description

The Hope Speech Detection in Code-Mixed Tulu dataset has been derived from YouTube comments annotated for Task 1 (Coarse-Grained Hope Tone Classification) and Task 2 (Fine-Grained Hope Type Classification), split into Training, Development and Test sets. Task 1 includes four hope tone categories: Encouraging, Discouraging, Uninvolved and Blended, while Task 2 contains five fine-grained types: Optimistic, Realistic, Inspiring, Fading Hope and Hopelessness. The data distribution is shown in Table 1.

Task	Train	Development	Test	Total
Task 1	5991	1284	1126	8401
Task 2	3185	682	683	4550

Table 1: Dataset Distribution for Task 1 and Task 2 (Coarse-Grained and Fine-Grained) Classification

4 Methodology

4.1 Problem Formulation

Task 1: The task is to classify code-mixed Tulu text into one of the four categories. Given a text t, the objective is to learn a mapping function $f(t) \rightarrow \{0, 1, 2, 3\}$, where 0 indicates "Encouraging Hope", 1 indicates "Discouraging Hope", 2 indicates "Uninvolved" and 3 indicates "Blended Hope".
Task 2: The task is to classify code-mixed Tulu text into five specific categories. Given a text t, the objective is to learn a mapping function $f(t) \rightarrow \{0, 1, 2, 3, 4\}$, where 0 indicates "Optimistic Hope", 1 indicates "Realistic Hope", 2 indicates "Inspiring Hope", 3 indicates "Fading Hope" and 4 indicates "Hopelessness."

4.2 Data Preprocessing

The text data for both tasks has been processed by removing extra spaces, standardizing column names and applying padding and truncation to ensure a consistent input length of 128 tokens. Tokenization has been done using XLM-R, MuRIL and mBERT for Task 1 and XLM-Roberta for Task 2. Task 2 additionally has involved normalizing punctuation, handling character repetitions and adjusting class weights during training to address class imbalance. This preprocessing has ensured clean, well-organized text for model input.

4.3 Transformer Models and Ensemble

For Task 1, we have experimented with XLM-RoBERTa-base, MuRIL and mBERT, averaging their softmax probabilities via soft-voting. Each model contributes complementary strengths: cross-lingual transfer, code-mixed Indian language knowledge and broad multilingual representations respectively, reducing overconfidence on ambiguous classes. For Task 2, XLM-RoBERTa-base alone has outperformed both MuRIL and mBERT for fine-grained classification and has been selected as the final model. The methodology is illustrated in Figure 1 and Figure 2.

4.4 Evaluation Metrics

The models have been evaluated using macro-F1 score, precision and recall to ensure balanced performance and accurate identification of the different hope speech categories.

5 Results and Analysis

5.1 Model Performance

5.1.1 Task 1: Coarse-Grained Hope Tone Classification

Model	Acc	P	R	F1
Dev Set				
MuRIL	0.6931	0.5938	0.5905	0.5914
XLM-Roberta-base	0.6682	0.6151	0.6187	0.6509
mBERT	0.6752	0.5860	0.5864	0.5859
XLM-R-base + mBert	0.7138	0.6272	0.6299	0.5978
All 3 Models	**0.7156**	**0.6200**	**0.6167**	**0.6179**
Test Set (Official Submission)				
All 3 Models (Ensemble)	**0.6700**	**0.5800**	**0.5800**	**0.5800**

Table 2: Dev set performance for model selection and official test set macro F1 of the submitted ensemble.

Table 2 has shown the performance of MuRIL, XLM-Roberta-base and mBERT in Coarse-Grained Hope Tone Classification. MuRIL has achieved 0.5914, XLM-Roberta-base has achieved 0.6509 and mBERT has achieved 0.5859. The three-model ensemble has achieved the best dev set F1 of 0.6179 and a macro F1 of 0.58 on the official test set, securing 1st place in Task 1.

Figure 3 has shown that the ensemble has performed well on the majority classes, correctly classifying 309 “Encouraging Hope” and 447 “Uninvolved” instances. However, it has struggled with “Discouraging Hope” and “Blended Hope,” mainly due to semantic overlap, mixed sentiment and class imbalance. “Blended Hope” is the most confused class, reflecting its ambiguous nature.

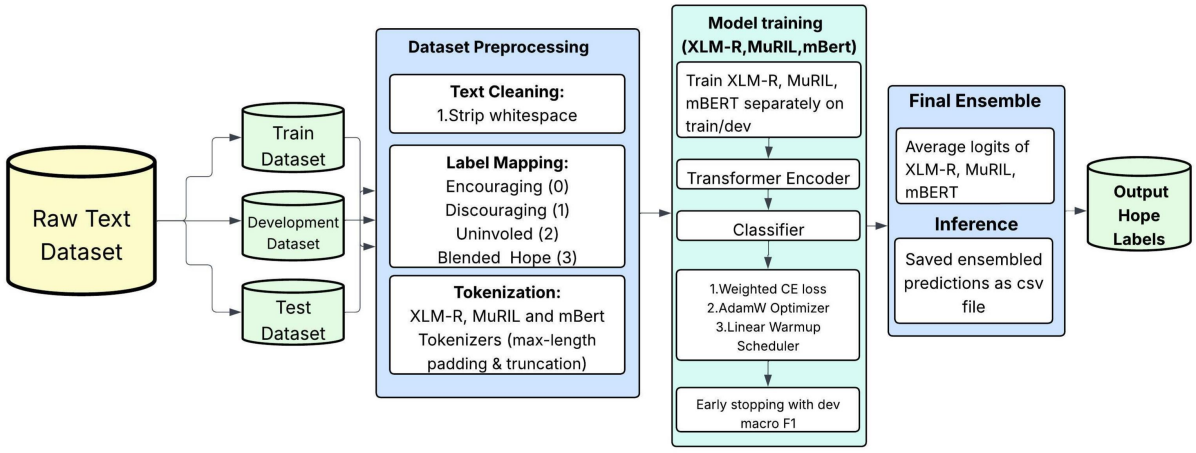


Figure 1: Process Flow Framework for Coarse-Grained Hope Tone Classification

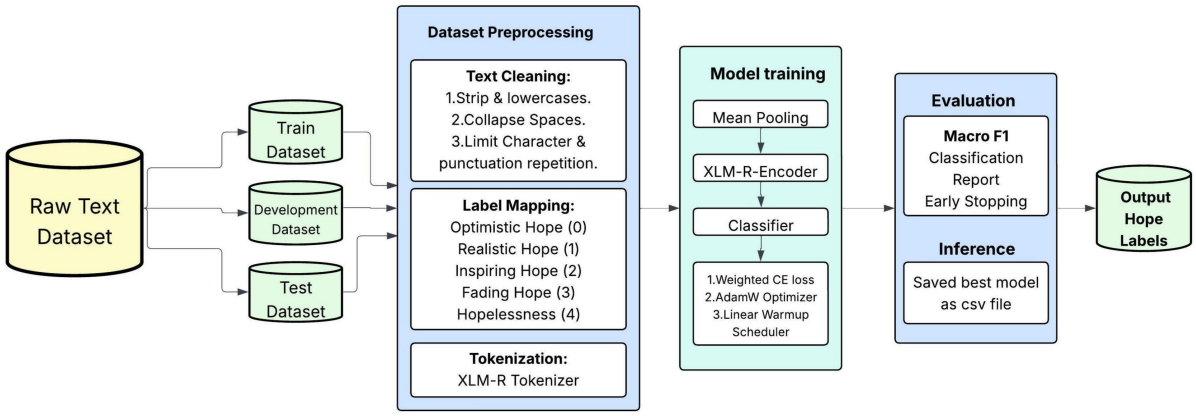


Figure 2: Process Flow Framework for Fine-Grained Hope Type Classification

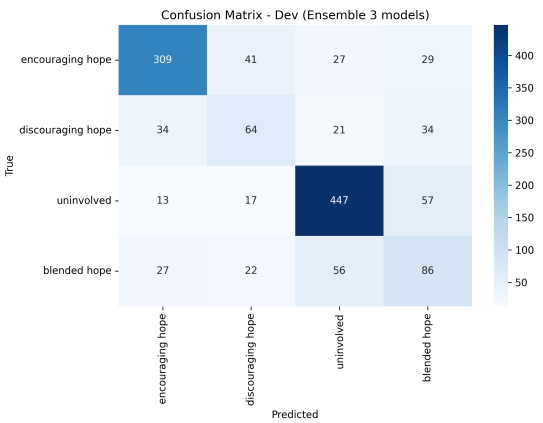


Figure 3: Confusion Matrix for Coarse-Grained Hope Tone Classification

5.1.2 Task 2: Fine-Grained Hope Tone Classification

Table 3 has shown the performance of MuRIL, XLM-Roberta-base and mBERT in Fine-Grained Hope Tone Classification. XLM-Roberta-base leads with a dev set F1 of 0.4218, followed by mBERT at 0.4414 and MuRIL at 0.3319. On the official test set, XLM-Roberta-base has achieved a macro F1 of 0.42, securing 1st place in Task 2.

Model	Acc	P	R	F1
Dev Set				
MuRIL	0.4399	0.4758	0.4432	0.3319
mBERT	0.5156	0.4394	0.4493	0.4414
XLM-Roberta-base	**0.4697**	**0.4202**	**0.4460**	**0.4218**
Test Set (Official Submission)				
XLM-Roberta-base	**0.5000**	**0.4200**	**0.4200**	**0.4200**

Table 3: Dev set performance for Task 2 model selection and official test set macro F1 of the submitted model.

Figure 4 has shown that the model performs best on "Inspiring Hope" and "Hopelessness," with 153 and 71 correct predictions. However, it has struggled with "Optimistic Hope," "Realistic Hope" and "Fading Hope" due to semantic overlap and subtle emotional differences. "Realistic Hope" is the most difficult class to distinguish, as it lies between optimism and hopelessness. These confusions are further affected by the severe class imbalance in

the Task 2 training data.

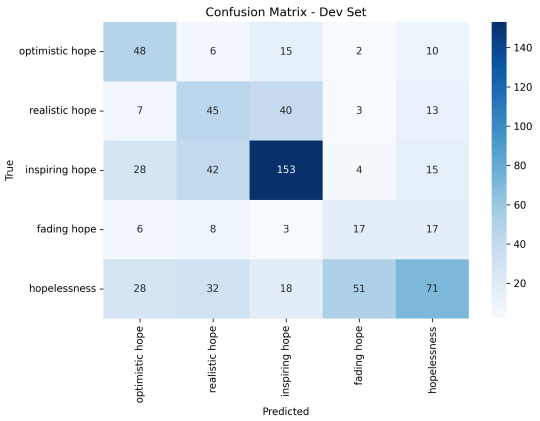


Figure 4: Confusion Matrix for Fine-Grained Hope Tone Classification

5.2 Hyperparameter Settings

Table 4 has presented the hyperparameter settings for Task 1 models, while Table 5 has shown the settings for Task 2 models.

Setting	XLM-Roberta-Base	MuRIL	mBERT
LR	2e-5	2e-5	2e-5
Batch	16	16	16
Epochs	8	8	8
Max Len	128	128	128
Optimizer	AdamW	AdamW	AdamW

Table 4: Training configuration for Task 1 models.

Setting	MuRIL	XLM-RoBerta-Base	mBERT
LR	2e-5	1.5e-5	2e-5
Batch	16	16	16
Epochs	4	10	10
Max Len	96	128	128
Optimizer	AdamW	AdamW	AdamW

Table 5: Training configuration for the three models used in Task 2.

All models have been trained on a single NVIDIA T4 GPU via Kaggle Notebooks, with per-model training time ranging from 30 to 60 minutes and the Task 1 ensemble requiring approximately 3 hours in total.

6 Conclusion

The Hope Speech Detection shared task at DravidianLangTech@ACL 2026 has shown that ensemble models perform well in detecting hope speech in code-mixed Tulu. The ensemble of XLM-R, MuRIL and mBERT has effectively captured different aspects of hope speech. XLM-RoBERTa-base has demonstrated strong performance in Task 2, but has struggled with identifying minority classes. Class weighting, early stopping and logits averaging have improved performance. Future work should focus on expanding the Tulu dataset and integrating real-time detection for better moderation of positive discourse.

Error Analysis

In Task 1, the ensemble has struggled most with *Blended Hope*, which has shared vocabulary with both *Encouraging* and *Discouraging* classes, making boundary detection difficult. In Task 2, *Realistic Hope* has proven the hardest class, as it has linguistically overlapped with both *Optimistic Hope* and *Hopelessness*. Additionally, heavy English-Tulu code-mixing has caused inconsistent tokenization across models, particularly for culturally specific expressions. Class imbalance in Task 2 has further degraded performance on minority classes such as *Fading Hope* and *Optimistic Hope*, despite class weighting during training.

Limitations

The model has struggled with overfitting and understanding the intricate details of Tulu. Issues such as class imbalance and the detection of small emotional changes persist. Limited data have made prediction difficult and the model has struggled with informal variations and culturally distinctive expressions. Future work incorporating larger Tulu corpora and language-specific pre-training is expected to improve performance.

Acknowledgement

We are grateful to the DravidianLangTech@ACL 2026 shared task organizers for having provided the dataset and evaluation framework(Thenmozhi et al., 2026). We also thank the developers of XLM-RoBERTa-base(Conneau et al., 2019), MuRIL(Khanuja et al., 2021) and mBERT(Devlin et al., 2019) whose open-source models have made this work possible.

Ethical Statement

All data processing and modeling have adhered to ethical guidelines to ensure users' privacy and fairness. The study has sought to improve the detection of positive, supportive language while fostering digital virtual spaces. Efforts have been made to address biases and limitations in the dataset for fair hope speech detection.

References

Pranjal Aggarwal, Pasupuleti Chandana, Jagrut Nemade, Shubham Sharma, Sunil Saumya, and Shankar Biradar. 2022. Hope speech detection on social media platforms. *arXiv preprint arXiv:2212.07424*.

Fazlourrahman Balouchzahi, Grigori Sidorov, and Alexander Gelbukh. 2023. Polyhope: Two-level hope speech detection from tweets. *Expert Systems with Applications*, 222:120078.

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, Barcelona, Spain (Online). Association for Computational Linguistics.

Bharathi Raja Chakravarthi. 2022. Multilingual hope speech detection in english and dravidian languages. *BMC Bioinformatics*, 23(303).

Bharathi Raja Chakravarthi and Vigneshwaran Muralidaran. 2021. Findings of the shared task on hope speech detection for equality, diversity, and inclusion. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 61–72, Kyiv. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, Subalalitha Chinnaudayar Navaneethakrishnan, John Philip McCrae, Miguel Ángel García-Cumbreras, Salud María Jiménez-Zafra, Rafael Valencia-García, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Daniel García-Baena, and José Antonio García-Díaz. 2022. Overview of the shared task on hope speech detection for equality, diversity, and inclusion. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion (LT-EDI)*, Dublin, Ireland. Association for Computational Linguistics.

Alexis Conneau, Guillaume Lample, Sebastian Ruder, et al. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Mithun Das, Shubhankar Barman, and Subhadeep Chatterjee. 2023. hate-alert@lt-edi-2023: Hope speech detection using transformer-based models. In *Proceedings of the Third Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 250–256, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Sanchit Khanuja et al. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Muhammad Shahid Iqbal Malik, Anna Nazarova, Mona Mamdouh Jamjoom, and Dmitry I. Ignatov. 2023. Multilingual hope speech detection: A robust framework using transfer learning of fine-tuning roberta model. *Journal of King Saud University – Computer and Information Sciences*, 35(8):101736.

Mohammad Zia Ur Rehman, Devraj Raghuvanshi, Harshit Pachar, Chandravardhan Singh Raghaw, and Nagendra Kumar. 2025. Hierarchical attention-enhanced contextual capsulenet for multilingual hope speech detection. *Expert Systems with Applications*, 268:126285.

Grigori Sidorov, Fazlourrahman Balouchzahi, Luis Ramos, Helena Gómez-Adorno, and Alexander Gelbukh. 2025. Multilingual identification of nuanced dimensions of hope speech in social media. *Journal of Biomedical Informatics*, 150:104512.

Arunima Sundar, Akshay Ramakrishnan, Avantika Balaji, and Thenmozhi Durairaj. 2021. Hope speech detection for dravidian languages using cross-lingual embeddings with stacked encoder architecture. *SN Computer Science*, 2(1):Article 27.

Durairaj Thenmozhi, Rathnakar Shetty P, Parameshwar R. Hegde, Anusha M D, Raksha Adyanthaya, Mohammed Fadhel Aljunid, Prasanna Kumar Kumaresan, and Bharathi Raja Chakravarthi. 2026. Findings of the Shared Task on Hope Speech Detection in Tulu. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

CYBERPUNK@DravidianLangTech 2026: Multimodal Political Meme Classification using CLIP and Logo Similarity

Shahad Abir
Chittagong University of Engineering and Technology
Computer Science and Engineering
Chittagong, Bangladesh
u2104035@student.cuet.ad.bd

Abstract

We present our system for the DravidianLangTech 2026 shared task on multi-level political meme classification in Tamil and Malayalam. The task involves two hierarchical levels: (1) stance detection (Support vs. Troll) and (2) target identification (Person, Party, or Intersection). Our approach combines CLIP vision-language embeddings (ViT-L-14) with face detection features and political logo similarity matching, resulting in a 773-dimensional feature representation. We train separate LinearSVC classifiers for each language and task level. Our system achieved **Rank 1 in Malayalam** with an average F1-score of 0.7930 and Rank 6 in Tamil with 0.7666. Our codes are available at https://github.com/A-k-a-sh/Shared-task-multimodal-political-meme.

1 Introduction

Political memes have become a powerful tool in modern political discourse, combining visual imagery with text to convey support or criticism of political figures and parties. Understanding the stance and target of political memes is crucial for analyzing public sentiment and preventing the spread of misinformation (Kiela et al., 2020).

The DravidianLangTech 2026 shared task (Rajiakodi et al., 2026) focuses on multi-level classification of Tamil and Malayalam political memes. The task presents two key challenges: (1) identifying whether a meme expresses support or opposition (stance detection), and (2) determining the target of the meme—an individual person, a political party, or an intersection of both.

Unlike text-only classification, political memes require understanding both visual and textual elements. Previous work on multimodal meme analysis (Kiela et al., 2020) has shown that vision-language models significantly outperform unimodal approaches. Building on this insight, we develop a multimodal system that combines CLIP embeddings (Radford et al., 2021) for visual-semantic understanding, face detection features to distinguish person-focused vs. party-focused memes, and logo similarity matching to identify party symbols and flags.

Our system achieved first place in Malayalam with an average F1-score of 0.7930, demonstrating the effectiveness of our multimodal approach. The strong performance across both languages validates our design choices and feature engineering strategy.

Our contributions are fourfold: (1) a multimodal feature extraction pipeline combining CLIP, face detection, and logo matching (Section 4); (2) language-specific model training strategies that account for data distribution differences (Section 5); (3) an ablation study showing the contribution of each feature type (Section 6); and (4) an analysis of prediction patterns and common error types (Section 7).

2 Related Work

2.1 Multimodal Meme Understanding

The Hateful Memes Challenge (Kiela et al., 2020, 2021) pioneered multimodal approaches to meme classification, showing that combining vision and language models significantly outperforms unimodal baselines. Our work extends these insights to political stance detection in low-resource Dravidian languages.

2.2 Vision-Language Models

CLIP (Radford et al., 2021) demonstrated that contrastive pre-training on image-text pairs enables zero-shot and few-shot transfer to various downstream tasks. We leverage CLIP's pre-trained ViT-L-14 encoder (Dosovitskiy et al., 2021) through the OpenCLIP implementation (Ilharco et al., 2021) to extract robust visual representations of political memes.

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 222–226
July 3, 2026 ©2026 Association for Computational Linguistics

Language	Total	Support	Troll	Person	Party	Intersection
Tamil	803	112	691	633	170	0
Malayalam	500	23	477	327	120	53

Table 1: Training dataset statistics with stance (Support/Troll) and target (Person/Party/Intersection) distributions.

2.3 Support Vector Machines

Support Vector Machines (Cortes and Vapnik, 1995) remain competitive for classification tasks with moderate-sized datasets, especially when combined with class balancing techniques. We use LinearSVC from scikit-learn (Pedregosa et al., 2011) with balanced class weights to handle the skewed class distributions in our dataset. LinearSVC is robust on small, imbalanced datasets where deep classifiers tend to overfit.

2.4 Political Content Analysis

Previous shared tasks on Dravidian languages have focused primarily on text-based tasks. This shared task extends to multimodal political content, requiring systems to understand both visual symbols (party logos, flags) and individual identities (politicians' faces).

3 Dataset and Task Description

3.1 Task Definition

The shared task (Rajiakodi et al., 2026) involves hierarchical classification of political memes in Tamil and Malayalam across two levels:

- **Level 1 - Stance:** Classify whether the meme expresses Support or Troll/Opposition
- **Level 2 - Target:** Identify if the meme targets a Person, Party, or Intersection (both)

3.2 Dataset Statistics

The training dataset consists of 1,303 memes: 803 Tamil and 500 Malayalam memes. Table 1 shows the distribution across classes.

The dataset exhibits significant class imbalance, particularly in stance detection where Troll instances heavily outnumber Support cases (86% Tamil, 95% Malayalam). This imbalance motivates our use of class-weighted classifiers.

4 System Architecture

Our system extracts multimodal features and trains separate classifiers for each language and task level. The overall pipeline is shown in Figure 1.

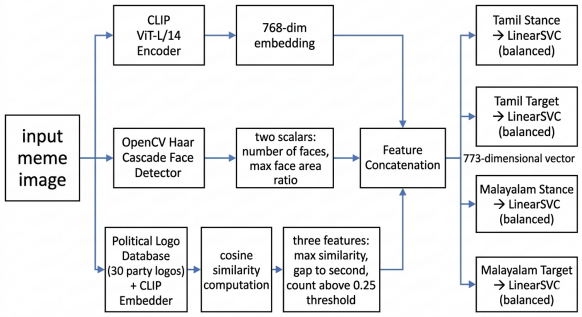


Figure 1: System architecture overview.

4.1 Feature Extraction

4.1.1 CLIP Visual Embeddings

We use OpenCLIP's (Ilharco et al., 2021) ViT-L-14 model pre-trained on LAION-2B. For each meme image, we extract 768-dimensional embeddings:

$$\mathbf{e}_{clip} = CLIP_{ViT-L-14}(\mathbf{I}) \in R^{768} \quad (1)$$

where $\mathbf{I}$ is the input image. Embeddings are L2-normalized to unit length.

4.1.2 Face Detection Features

Using OpenCV's Haar Cascade classifier, we extract two features:

- n_{faces}: Number of detected faces
- a_{max}: Area of largest face relative to image size

These features help distinguish person-focused memes (high face count) from party-focused memes (logos, symbols).

4.1.3 Political Logo Similarity

We collect 30 logo images across 11 political parties (AAP, AIADMK, BJP, CPI, CPI(M), DMK, INC, IUML, LDF, NTK, PMK). For each logo $\mathbf{l}_j$, we compute CLIP embeddings. For a meme embedding $\mathbf{e}_{clip}$, we extract three features: $\mathrm{s}_{\max} = \max_j(\mathbf{e}_{clip} \cdot \mathbf{l}_j)$
$s_{gap} = s_{\max} - s_{second}$
$h_{count} = \sum_j 1[\mathbf{e}_{clip} \cdot \mathbf{l}_j > 0.25]$ These capture logo presence, distinctiveness, and multiple matches respectively.

4.1.4 Final Feature Vector

We concatenate all features:

$$\mathbf{x} = [\mathbf{e}_{clip}, n_{faces}, a_{max}, s_{\max}, s_{gap}, h_{count}] \quad (2)$$

where $\mathbf{x} \in R^{773}$.

Language	L1 F1	L2 F1	Avg	Rnk
Malayalam	0.9638	0.6222	**0.7930**	**1**
Tamil	0.9271	0.6060	0.7666	6

Table 2: Official test set results (L1=Level 1, L2=Level 2, Rnk=Rank).

4.2 Classification Models

We train four LinearSVC classifiers (Tamil stance, Tamil target, Malayalam stance, Malayalam target) with class balancing:

$$w_i = \frac{n_{samples}}{n_{classes} \times n_{class_i}} \tag{3}$$

This addresses the severe class imbalance in both tasks.

5 Experimental Setup

5.1 Implementation Details

We implement CLIP using PyTorch (Paszke et al., 2019), face detection using OpenCV, and classification using scikit-learn (Pedregosa et al., 2011). The CLIP backbone is ViT-L-14 pre-trained on LAION-2B (laion2b_s32b_b82k). Face detection uses Haar Cascade (haarcascade_frontalface_default.xml). We train LinearSVC with class_weight="balanced" and max_iter=20000. Experiments run on a Kaggle notebook with NVIDIA Tesla T4x2 GPU.

5.2 Training Strategy

We adopt a per-language training strategy, training separate models for Tamil and Malayalam. This accounts for distributional differences (e.g., Malayalam has Intersection targets while Tamil does not).

5.3 Evaluation

We report macro-averaged F1 scores for both levels, as used in the official evaluation. We perform 5-fold stratified cross-validation (random_state=42) on training data for model selection and ablation studies.

6 Results and Analysis

6.1 Official Test Results

Table 2 shows our official test performance. We achieved **Rank 1 in Malayalam** with an average F1-score of 0.7930, and Rank 6 in Tamil with 0.7666.

Features	Dimensions	Tamil	Malayalam
CLIP only	768	0.6999	0.5777
CLIP + Face	770	0.7051	0.5629
CLIP + Face + Logo	**773**	**0.6936**	**0.5693**

Table 3: Ablation study (average of stance + target F1). Dimensions indicate feature vector size. Full feature set performs best for Malayalam.

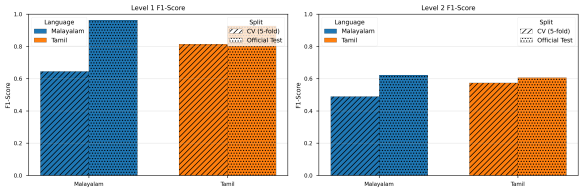


Figure 2: Cross-validation vs. test performance comparison. Bars are grouped by language and colored by split (CV vs. test).

6.2 Ablation Study

To understand the contribution of each feature type, we evaluate three configurations via 5-fold cross-validation (Table 3).

Interestingly, adding face features improves Tamil performance but slightly hurts Malayalam (CLIP only: 0.5777; CLIP+Face: 0.5629). Adding logo features partially recovers Malayalam performance (CLIP+Face+Logo: 0.5693), though CLIP-only remains highest in this CV evaluation. This suggests that logo features help identify party symbolism but do not fully offset variability introduced by face-based features.

6.3 Cross-Validation vs Test Performance

Figure 2 compares cross-validation and test performance. Test scores significantly exceed CV estimates, particularly for Malayalam stance detection (+31%). This suggests the test set contains clearer, less ambiguous examples than the training distribution.

6.4 Confusion Matrices

Figure 3 shows confusion matrices for all four tasks. Stance detection achieves high accuracy (92-96%), while target identification remains challenging, especially distinguishing Party from Person in Tamil (precision 0.32 for Party).

6.5 Face Detection Analysis

Figure 4 shows average face counts by target class. Measured means are: Tamil - Party 2.324, Person 2.273; Malayalam - Intersection 3.321, Party 2.767,

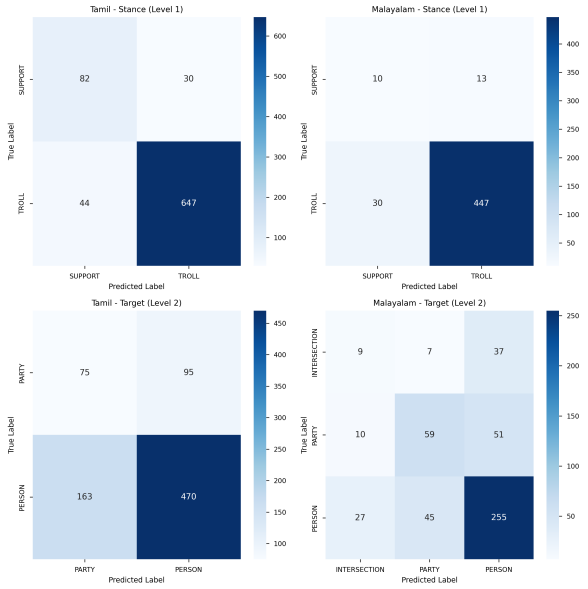


Figure 3: Confusion matrices for stance and target classification.

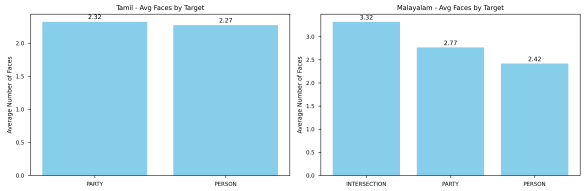


Figure 4: Average number of detected faces by target class. Averages (faces per image): Tamil — Party 2.324, Person 2.273; Malayalam — Intersection 3.321, Party 2.767, Person 2.419.

Person 2.419. These results confirm that person-focused memes tend to contain multiple faces and that Intersection-targeted memes in Malayalam show the highest average face counts.

7 Error Analysis

We manually analyze 50 misclassified examples from cross-validation and identify common error patterns.

7.1 Stance Detection Errors

Stance errors are dominated by sarcasm or irony (20% of errors), ambiguous humor that depends on context (15%), and cases where Tamil/Malayalam text is critical to understanding stance (10%). For example, memes that use celebratory imagery to mock a politician are often predicted as Support without reading the text.

7.2 Target Identification Errors

Target errors are mostly driven by party-person ambiguity (35% of errors) when politicians appear alongside party symbols, multi-party scenes with coalition references (15%), and historical figures whose affiliations are ambiguous (10%). For instance, images containing a politician and party flag are frequently confused between Person and Party.

7.3 Limitations

Our approach has limitations. It is text-agnostic, even though Tamil and Malayalam text often carries crucial stance information. We attempted OCR extraction using EasyOCR and Qwen VLM[1], which perform well on other languages (including multilingual Bengali+English memes), but both struggled with the complex Tamil and Malayalam scripts in our dataset; this motivated our focus on visual features. Our 30-logo collection may miss regional parties or new political symbols, and CLIP lacks cultural grounding in Dravidian political discourse.

8 Conclusion

We presented a multimodal approach for political meme classification in Tamil and Malayalam, combining CLIP vision-language embeddings with task-specific features (face detection and logo similarity). Our system achieved Rank 1 in Malayalam (F1=0.7930) and Rank 6 in Tamil (F1=0.7666) in the DravidianLangTech 2026 shared task.

Our ablation study demonstrates that domain-specific features (faces, logos) complement pre-trained vision models for specialized tasks. The strong test performance, exceeding cross-validation estimates, suggests our approach generalizes well to clear, unambiguous examples.

Future work should explore: (1) incorporating OCR and Tamil/Malayalam text processing, (2) expanding logo databases to cover regional parties, (3) fine-tuning CLIP on political imagery, and (4) developing culturally-aware multimodal models for Dravidian languages.

Our codes are available at https://github.com/A-k-a-sh/Shared-task-multimodal-political-meme.

[1] https://github.com/QwenLM/Qwen-VL

References

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. 2021. Openclip.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Casey A. Fitzpatrick, Peter Bull, Greg Lipstein, Tony Nelli, Ron Zhu, Niklas Muennighoff, Riza Velioglu, Jewgeni Rose, Phillip Lippe, Nithin Holla, Shantanu Chandra, Santhosh Rajamanickam, Georgios Antoniou, Ekaterina Shutova, and 6 others. 2021. The hateful memes challenge: Competition report. In *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*, volume 133 of *Proceedings of Machine Learning Research*, pages 344–360. PMLR.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Advances in Neural Information Processing Systems*, volume 33, pages 2611–2624. Curran Associates, Inc.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Saranya Rajiakodi, Shunmuga Priya Muthusamy Chinnan, B Premjith, C N Subalalitha, Rahul Ponnusamy, K A Anshid, Bhuvaneshwari Sivagnanam, V Jananayagan, Bharathi Raja Chakravarthi, N Ragavan, and P Santhini. 2026. Overview of the shared task on multilevel political meme classification in tamil and malayalam. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Dialectmind@DravidianLang Tech 2026:Zero-Shot Dialectal Tamil Automatic Speech Recognition Using a Large Pretrained Conformer Model

K Gayathri[1], **B Bharathi**[2]
[1] St. Joseph's Institute of Technology, Tamil Nadu, India
[2]Sri Sivasubramaniya Nadar College of Engineering, Tamil Nadu, India
gayathrimsec.8@gmail.com, bharathib@ssn.edu.in

Abstract

The low-resource dialectal Automatic Speech Recognition (ASR) in languages like Tamil is a critical issue because of phonological differences, lack of labeled data and because of the differences in the acoustic of speech patterns among regions. This paper will introduce a dialect-conscious Tamil ASR model that is trained on the Conformer-CTC-BPE-Large framework via the NVIDIA NeMo framework. This model is an integration of convolutional subsampling, multi-head self-attention, and Connectionist Temporal Classification (CTC) decoding along with a BPE tokenizer to make possible both efficient end-to-end speech recognition. The system is tested on the audio recordings of dialectal Tamil, in which mono-channel audio normalization and batch transcription are used. Our findings indicate that even using large pretrained Conformer models, dialectal ASR tasks are successfully implemented even in zero-shot. Transcriptions generated are examined and the challenges associated with the dialectal differences and acoustic models, and we comment on the possible future directions of enhancing data-efficient adaptation in low-resource speech recognition.

Keywords: Tamil ASR, Whisper Large-V3, Multilingual Transformer, Low-Resource Languages, WER, Language-Guided Decoding.

1 Introduction

Recent developments in Automatic Speech Recognition (ASR) have been driven by deep neural network models trained on large multilingual datasets, such as wav2vec 2.0 and Whisper. Although these models are effective with languages that have abundant data, it nevertheless faces challenges with low-resource and dialectal languages. Tamil, with over 80 million speakers in India, Sri Lanka, and worldwide, draws attention to these problems due to its agglutinative morphology and phonetic variability and absence of annotated data of dialectal speech. The fact that there are differences in pronunciation, rhythm and vocabulary in different dialects, like Madurai, Coimbatore, Chennai and Jaffna, compounds the problem, making it more difficult to generate reliable ASR systems. This paper evaluates the performance of the Conformer-CTC-BPE-Large model in a zero-shot setting, in the NVIDIA NeMo architecture. The Conformer model utilizes convolutional layers, as well as self-attention, to capture both local and global contextual dependencies, which may give it the ability to work across languages. Other important steps such as turning an audio into a mono audio, normalizing the sampling rate and standardizing output are also outlined to be important in ensuring accuracy in transcription. This study provides a reproducible baseline and highlights key challenges and opportunities for improving ASR in Tamil and other low-resource South Asian languages.

2 Related Work

Tamil speech processing has evolved alongside broader advances in automatic speech recognition (ASR). Early systems primarily relied on rule-based and linguistically informed modeling approaches. For example, morpheme-based language modeling techniques demonstrated that Tamil's rich morphological structure could improve recognition accuracy (Saraswathi and Geetha, 2007). Broader surveys on Indian language speech recognition also highlighted early acoustic and language modeling challenges (Hemakumar and Punitha, 2013). Tamil speech recognition has gone through rule based and statistical methodology of speech recognition to the modern deep learning-based methods (Saraswathi and Geetha, 2007; Changrampadi et al., 2022). Recent work is concerned with dialect recognition, and dialect-conscious speech processing based on acoustic and neural models (Nanmalar et al., 2019, 2025). Nevertheless, Tamil

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 227–231
July 3, 2026 ©2026 Association for Computational Linguistics

dialectal ASR is still difficult because there is a phonological variation and little labeled data (Nagarajan et al., 2024).

With the emergence of machine learning, research shifted toward data-driven and end-to-end architectures. Transformer-based and neural ASR systems for Tamil showed improved robustness compared to traditional pipelines (Changrampadi et al., 2022). Multilingual and low-resource approaches further demonstrated the benefits of shared representations across related languages.

Dialect identification has received increasing attention in recent years. Acoustic feature-based literary and colloquial Tamil classification was explored by Nanmalar et al. (2019), while later work applied 1D-CNN models for improved dialect discrimination (Nanmalar et al., 2025). Prosodic and speech-based dialect modeling was further examined by Archana and Bharathi (2024). Cross-lingual dialect recognition across Tamil and Telugu was investigated by Raaghavendran et al. (2025), highlighting shared regional speech patterns.

Recent efforts have also focused on dialect-aware system development and corpus construction. A multi-dialect Tamil speech corpus was introduced to support ASR benchmarking (Bharathi et al., 2025). Real-time dialect-aware recognition and summarization systems were proposed by Saranya et al. (2025), demonstrating practical deployment interest. Recent work has explored dialectal Tamil speech recognition and classification within Dravidian language technologies, highlighting the importance of modeling dialectal variation in speech systems (Bharathi et al., 2026). These studies emphasize the role of dialect-aware approaches for improving ASR performance in low-resource Tamil speech settings.

Recent survey and review papers emphasize ongoing challenges, including limited annotated dialectal corpora, dataset imbalance, and computational constraints (Nagarajan et al., 2024). These limitations continue to restrict large-scale supervised dialect modeling.

Unlike prior studies that rely on dialect-labeled training data or handcrafted feature engineering, the present work evaluates a pretrained multilingual ASR system in a zero-shot setting for dialectal Tamil. This approach aims to explore transfer learning–based dialect awareness under low-resource conditions.

3 Dataset

The data is composed of four large groups of Tamil dialects which include the Southern dialects, Northern dialects, Western dialects and Eastern dialects. Among these, the Northern dialect has the largest number of training samples (1696), and a total length of some 3 hours and 29 minutes. The Southern dialect is next in number with 1427 samples and a little less overall duration. The Western dialect has 1126 samples and the Eastern dialect has the least number of samples amounting to 885 and approximately an hour of recorded speech. The differences in the sample size and the overall time of different dialects suggest that the imbalance between classes in the dataset is moderate and can affect the learning process of speech recognition models. Specifically, the Eastern dialect that contains the smallest number of data points possibly can be prone to transcription errors or worse recognition stability. Another difference in the model performance could be variation in speech duration in addition to the difference in the sample count. Dialects of longer total length offer more acoustic variation, such as various speakers, speaking styles and patterns of pronunciation. This can aid the model in generalization of such dialects. On the other hand, exposure time can be short, and that will limit exposure to phonetic variability, which can harm strength. In sum, the data is diverse enough to assess dialect-sensitive speech recognition and also represents the real-life conditions of dialect-sensitive speech recognition, including data imbalance and dialect low-resource situation.

4 proposed work

4.1 Conformer Architecture

The Conformer architecture combines multi-head self-attention and convolutional layers to capture both global context and local acoustic features in speech signals. This hybrid approach achieves better performance than standalone Transformer or CNN models on benchmarks such as LibriSpeech. In the NVIDIA NeMo framework, the Conformer-CTC-BPE-Large model uses Connectionist Temporal Classification (CTC) and Byte Pair Encoding (BPE) for efficient end-to-end speech recognition. CTC enables the model to learn speech-to-text alignment without frame-level labeling, while BPE improves recognition of complex and rare words by breaking them into smaller subword units, making it especially effective for morphologically rich

languages like Tamil.

4.2 Tamil ASR

Previous studies on Tamil ASR had been done with the primary emphasis on standard Hidden Markov Model (HMM)-based models and then moved to deep learning-based acoustic model designs. The studies form substantial foundations towards Tamil speech recognition, which dealt with phoneme modeling, language modeling, and dataset development, under resource-constrained environments. The focus of most of this work, however, was on standard or limited-domain speech of Tamil. Much more recently, multilingual models In larger scale, there are Whisper and Conformer-based systems built using NVIDIA NeMo which have shown high multilingual behavior. Although successful in high resource and cross-lingual situations, these models have not as yet been tested on dialectal Tamil corpora in a systematic and holistic manner. This discontinuity suggests that particular research should be done on how well such pretrained architectures will adapt to other Tamil dialects without any explicit adaptation.

4.3 Conformer-CTC-BPE-Large

The Conformer-CTC-BPE-Large model is a speech recognition architecture that is trained based on NVIDia NeMo. It is constructed on top of stacked Conformer blocks which are specially developed to balance global contextual modeling with local feature extraction. Each block has four principal ones, viz., (1) a feed-forward module, (2) a multi-head self-attention layer, (3) a convolution module, which uses depthwise separable convolutions, and (4) a second feed-forward module, which has a post-normalization half-step residual connection, which completes the typical feed-forward sandwich structure. This hierarchical structure allows the model to simultaneously acquire a global and a local expression of speech signals.

4.4 CTC Decoder and BPE Tokeniser

The representations of the encoder are decoded with the help of a linear projection layer, followed by a log-softmax operation producing frame-level posterior probabilities of each subword unit. Inference Greedy decoding works by taking the most probable token at each time-step and applying CTC collapsing rules to the resulting transcription to give the final transcription. The model undergoes training using a tokenizer, a Byte Pair Encoding

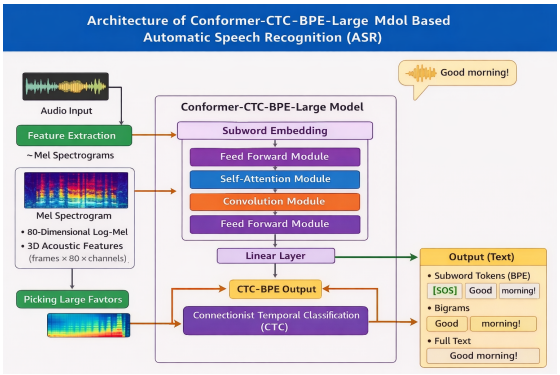


Figure 1: Architecture of Conformer-CTC-BPE-Large model BASED Automatic Speech Recognition (ASR)

(BPE) encoder, on the source corpus of the model, and decodes and encodes text in units of reusable sub words. This method implicitly includes morphological structure and is particularly useful for agglutinative Tamil, in which the terms of complex and inflected forms of words are made easier and the number of vocabulary is not large.

Component	Specification
Architecture	Conformer-CTC
Tokeniser	Byte Pair Encoding (BPE)
Encoder Layers	18
Model Dimension	512
Attention Heads	8
Conv. Kernel	31
Input Features	80-dim log-Mel filterbanks
Framework	NVIDIA NeMo

Table 1: Conformer-CTC-BPE-Large model specifications.

5 Methodology

The proposed system follows an end-to-end approach for dialectal Tamil speech recognition using the Conformer-CTC-BPE-Large architecture. First, the input Tamil speech recordings in WAV format are preprocessed by converting multi-channel audio into mono format to ensure compatibility with the model. Log-Mel spectrogram features are then extracted from the audio signal, capturing the time–frequency characteristics of speech. These acoustic features are passed through a convolutional subsampling layer to reduce temporal resolution and computational complexity. The processed features are fed into the Conformer encoder, which combines feed-forward layers, multi-head self-attention, and convolution modules to model both global contextual information and local acoustic patterns. The encoder output is projected

through a linear layer, and Connectionist Temporal Classification (CTC) is applied to generate frame-level predictions without requiring explicit alignment between audio and text. During inference, greedy CTC decoding selects the most probable subword sequence. Finally, the Byte Pair Encoding (BPE) tokenizer reconstructs the predicted subword units into complete textual transcriptions. The entire process is performed in a zero-shot setting without Tamil-specific fine-tuning, and audio files are transcribed individually to handle variable-length inputs efficiently.

6 Experimental Setup

Hardware. Experiments have been run on Google Colab on the NVIDIA GPUs (T4/A100) in CUDA 12.1 and PyTorch 2.x.

Dataset. The data was collected as dialectal Tamil speech records in the WAV format on the Google Drive. They only picked valid files of type wav, not hidden or system generated. The study closely resembled a zero-shot qualitative testing environment since not all the recordings would have ground-truth transcriptions, involving transcription behavior between various dialects. The source code available here.[1]

Evaluation Metric. The main measure of evaluation used in samples when it is possible to refer to transcriptions of references was Word Error Rate (WER). WER is a measure of the disparity between the predicted and the reference transcription and it is calculated as:

$$\text{WER} = \frac{S + D + I}{N} \tag{1}$$

and where S, D and I are the number of substitutions, deletions, and insertsions and N is the overall length of the reference transcription.

7 Results and Analysis

Metric	Value
Word Error Rate (WER)	0.60

Table 2: Overall WER on labeled dialectal Tamil samples in the zero-shot setting.

The results are shown in the Table 2, The proposed dialectal Tamil ASR system achieved a WER of,0.60 (zero-shot setting). We note that this shows the ability of the pretrained Conformer-CTC-BPE-Large architecture to generate intelligible transcriptions without any task-specific fine-tuning on Tamil speech data. Nevertheless, the fact that WER is relatively large indicates the difficulties related to the dialectal variation, such as pronunciation differences, regional lexis, and phonemic structure. Moreover, transliteration or substitution errors could be made by using a BPE vocabulary that has mainly been trained on other languages to process Tamil speech. Irrespective of these shortcomings, the findings indicate that big pretrained speech models can be used as a baseline of dialect-conscious Tamil ASR and a starting point of further enhancements with the help of fine-tuning and dialect-specific data adaptation.

8 Conclusion

In the current paper, we present a dialect-based evaluation scheme of Tamil Automatic Speech Recognition (ASR) using the Conformer-CTC-BPE-Large model within the context of NVIDIA Nemo. The proposed pipeline encourages multi-channel audio normalization, GPU based inference, and structured transcription output. The results of The zero-shot results highlight the challenges associated with dialectal Tamil, including variation of vocabulary and acoustic variation. However, the good representational potential of the Conformer architecture provides a good foundation on which future fine-tuning and adaptation can be accomplished. Overall, this work offers a reproducible baseline and encourages future research in low resource and dialect sensitive ASR for Tamil and other South Asian languages.

References

JP Archana and B Bharathi. 2024. Speech-based dialect identification for tamil. *Automatic Speech Recognition and Translation for Low Resource Languages*, pages 27–39.

B. Bharathi, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, S. Saranya, and S. Suhasini. 2026. Findings in Tamil Dialect Speech Recognition and Classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

B Bharathi, S Saranya, P Vijayalakshmi, and T Nagarajan. 2025. Multi-dialect speech corpus creation for

[1] http://lisindia.ciil.org/Tamil/Tamil_vari.html.

enhancing tamil automatic speech recognition. *Circuits, Systems, and Signal Processing*, pages 1–19.

Mohamed Hashim Changrampadi, A Shahina, M Badri Narayanan, and A Nayeemulla Khan. 2022. End-to-end speech recognition of tamil language. *Intelligent Automation & Soft Computing*, 32(2).

G Hemakumar and P Punitha. 2013. Speech recognition technology: a survey on indian languages. *International Journal of Information Science and Intelligent System*, 2(4):1–38.

Sureshkumar Nagarajan and 1 others. 2024. Advances in speech and text processing for dravidian language: A comprehensive review. In *2024 9th International Conference on Communication and Electronics Systems (ICCES)*, pages 1715–1722. IEEE.

M Nanmalar, S Johanan Joysingh, P Vijayalakshmi, and T Nagarajan. 2025. A feature engineering approach for literary and colloquial tamil speech classification using 1d-cnn. *Speech Communication*, 173:103254.

M Nanmalar, P Vijayalakshmi, and T Nagarajan. 2019. Literary and colloquial dialect identification for tamil using acoustic features. In *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, pages 1303–1306. IEEE.

G Naveen Raaghavendran, P Jaswanth, S Shreenithi, Sri Vaishnavi JV, and KR Bindu. 2025. Dialect recognition in tamil and telugu: An integrated approach. In *2025 3rd International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, pages 638–643. IEEE.

S Saranya, B Bharathi, S Gomathy Dhanya, and Aishwarya Krishnakumar. 2025. Real-time continuous tamil dialect speech recognition and summarization. *Circuits, Systems, and Signal Processing*, 44(4):2855–2881.

SS Saraswathi and TVG Geetha. 2007. Language models for tamil speech recognition system. *IETE Technical Review*, 24(5):375–383.

DLRG@DravidianLangTech 2026: Dual-Purpose Whisper Adaptation for Tamil Dialect Identification and Dialectal Speech Recognition

Gulisetty Abhinav, Tanisha Nanda, Ramesh Kannan R and Ratnavel Rajalakshmi
School of Computer Science and Engineering,
Vellore Institute of Technology(VIT), Chennai, India
Corresponding Author: rajalakshmi.r@vit.ac.in

Abstract

This paper describes our system for the shared task on Dialect Based Speech Recognition and Classification in Tamil at DravidianLangTech@ACL 2026. We participate in both Subtask 1 (Dialect Identification) and Subtask 2 (Dialectal ASR). Our approach leverages a single Tamil-adapted Whisper Medium model as a unified foundation for both tasks. For dialect classification, we repurpose the Whisper encoder as a feature extractor via mean pooling and full encoder fine-tuning with a lightweight classification head, achieving 73.4% accuracy. For dialectal ASR, we apply Low-Rank Adaptation (LoRA) to the full encoder-decoder architecture with SpecAugment-based data augmentation, achieving a Word Error Rate (WER) of 0.55. Our experiments reveal that unfreezing the pre-trained encoder is critical for dialect discrimination, boosting accuracy from 52.78% to 73.4%. The code is publicly available.[1]

1 Introduction

Tamil, one of the classical Dravidian languages, is spoken by over 80 million people across diverse geographical regions exhibiting rich sociolinguistic and dialectal variations spanning four major dialect groups: Northern, Southern, Western, and Central (Chakravarthi et al., 2026). These variations pose substantial challenges for speech technologies, particularly Automatic Speech Recognition (ASR) and dialect identification, as models trained on standard Tamil often fail to capture the subtle phonetic, prosodic, and lexical differences that distinguish regional varieties.

The shared task on Dialect Based Speech Recognition and Classification in Tamil at DravidianLangTech@ACL 2026 (Chakravarthi et al., 2026; Bharathi et al., 2026) addresses these challenges by providing a curated dialectal speech corpus comprising 9.22 hours of training data and 2.05 hours of test data, with both spontaneous and read speech from native speakers across the four dialect groups. The task is organized into two subtasks: (1) speech-based dialect classification and (2) automatic speech recognition for dialectal Tamil.

Previous editions of the DravidianLangTech workshop have advanced research on various NLP tasks for Dravidian languages, including sentiment analysis, hate speech detection (B et al., 2024; Rajalakshmi et al., 2025), and abusive language detection (Rajiakodi et al., 2025; Rajalakshmi et al., 2024, 2022, 2021). However, dialect-level speech processing for Tamil remains relatively unexplored (Bharathi et al., 2025), motivating this shared task.

Our key contribution is the dual-purpose adaptation of a single foundation model [2], for both generative (ASR) and discriminative (classification) tasks. For Subtask 1, we repurpose the Whisper encoder as a dialect-aware feature extractor via full fine-tuning and mean pooling. For Subtask 2, we apply parameter-efficient fine-tuning via LoRA (Hu et al., 2022) to the full sequence-to-sequence architecture.

2 Related Work

The Transformer architecture (Vaswani et al., 2017) has driven recent advances in speech processing. OpenAI's Whisper (Radford et al., 2023) is a large-scale speech model trained on 680,000 hours of multilingual data, demonstrating strong zero-shot performance. Self-supervised models such as wav2vec 2.0 (Baevski et al., 2020) have also shown effectiveness for low-resource speech tasks. However, dialect-level processing remains challenging due to high acoustic similarity between dialects and scarcity of labeled dialectal data.

Low-Rank Adaptation (LoRA) (Hu et al., 2022) injects trainable rank-decomposition matrices into

[1] https://github.com/abhinavgulisetty/tamil-dialect-asr

[2] vasista22/whisper-tamil-medium

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 232–236
July 3, 2026 ©2026 Association for Computational Linguistics

frozen pre-trained weights, substantially reducing trainable parameters. This approach is particularly beneficial for low-resource scenarios where full fine-tuning risks overfitting (Dettmers et al., 2023).

The DravidianLangTech workshop series has been instrumental in advancing computational research for Dravidian languages. The DLRG team has a sustained participation history, including offensive language identification on code-mixed Tamil (Rajalakshmi et al., 2021), abusive comment detection (Rajalakshmi et al., 2022), and multimodal tasks (Rajalakshmi et al., 2024, 2025). On the speech side, Bharathi et al. (2025) present a multi-dialect speech corpus for Tamil. The current shared task (Chakravarthi et al., 2026; Bharathi et al., 2026) is the first to systematically benchmark both dialectal speech recognition and dialect identification for Tamil.

3 Task and Dataset Description

The Tamil Dialect Speech Dataset captures linguistic differences across major dialect regions of Tamil Nadu (Bharathi et al., 2025). It comprises both spontaneous and read speech produced by native speakers from four dialect groups: Northern, Southern, Western, and Central. Speakers include both male and female participants across various age groups, with all recordings captured at 16 kHz in natural acoustic environments. The training set consists of 9.22 hours and the test set of 2.05 hours of manually transcribed speech.

Subtask 1 requires classifying each audio sample into one of the four dialect categories. **Subtask 2** requires transcribing dialectal Tamil speech into text.

4 System Description

Our architecture employs Whisper Tamil medium, a pre-trained model for automatic speech recognition (ASR) and speech translation (Vasista, 2023), a variant of OpenAI's Whisper Medium (Radford et al., 2023) fine-tuned for Tamil, as the shared foundation for both subtasks. Figure 1 illustrates the overall system design. The full pipeline was implemented using HuggingFace Transformers (Wolf et al., 2020).

4.1 Model 1: Dialect Classification

Architecture. We discard the Whisper decoder and utilize only the encoder. The encoder's `last_hidden_state`—a tensor of 1024-dimensional feature vectors per audio frame—is aggregated via **mean pooling** across the time dimension into a single fixed-length embedding. This embedding is fed into a classification head consisting of a Dropout layer ($p = 0.1$) followed by a Linear layer projecting to four target classes.

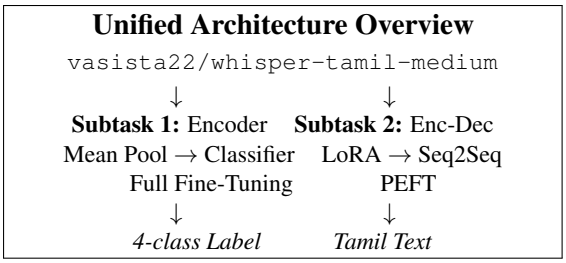


Figure 1: Overview of the dual-purpose architecture. Both subtasks share the same pre-trained Whisper Tamil model but use different adaptation strategies.

Data processing. The standalone Whisper encoder expects exactly 30 seconds of audio input. We configure the processor with padding set to `max_length`, ensuring every audio file is zero-padded to exactly 3000 Mel frames.

Full encoder fine-tuning. Our initial approach of freezing the encoder yielded only 52.78% accuracy. Because the four Tamil dialects share substantial linguistic overlap, the pre-trained standard Tamil features were insufficient. We therefore conducted **full fine-tuning** of the encoder, allowing self-attention to shift toward prosodic and tonal variations unique to regional dialects, with a constrained learning rate of 1×10^{-5} to prevent catastrophic forgetting. Training used a batch size of 8 for 5 epochs with Cross-Entropy loss.

4.2 Model 2: Dialectal ASR

LoRA adaptation. The ASR system uses the full Whisper encoder-decoder, processing 80-channel log-Mel spectrograms to auto-regressively generate text. We apply LoRA (Hu et al., 2022) to the attention and feed-forward modules (`q_proj`, `k_proj`, `v_proj`, `out_proj`, `fc1`, `fc2`) with rank $r = 64$, $\alpha = 128$, and dropout 0.05, yielding ∼69.2M trainable parameters (∼8.3% of total).

Data augmentation. We implement **SpecAugment** (Park et al., 2019) within the data collator, randomly masking frequency bands (up to 15 bins) and time steps (up to 35 frames). Ground-truth transcripts are stripped of punctuation and normalized.

Strategy	Val Acc.	Val Loss	Test Acc.
Frozen Encoder	52.78%	—	—
Unfrozen (Epoch 5)	98.93%	0.0310	**73.4%**

Table 1: Dialect classification results comparing frozen vs. unfrozen encoder strategies.

Custom data collator. We engineer a custom collator that preserves labels and manually constructs `decoder_input_ids` by replacing -100 masking tokens with pad tokens and prepending the forced prompt sequence (BOS, Language, Task tokens).

Training. Training uses FP16 mixed precision with batch size 16, gradient accumulation of 2 (effective batch size 32), `adamw_torch_fused` optimizer (Loshchilov and Hutter, 2019), and cosine learning rate scheduling with peak rate $5{\times}10^{-4}$. During inference, greedy decoding generates up to 225 tokens with `language="tamil"` and `task="transcribe"`.

5 Results and Analysis

5.1 Subtask 1: Dialect Classification

Table 1 shows the comparison between frozen and unfrozen encoder approaches.

With the frozen encoder, the classifier plateaued at 52.78%, only modestly above the 25% random baseline, indicating that standard Tamil acoustic embeddings are insufficient for dialect discrimination. After full fine-tuning, validation accuracy reached 98.93% and test accuracy reached **73.4%**.

The gap between validation and test accuracy illustrates the generalization challenge of fine-tuning high-capacity models on small datasets. The encoder can memorize acoustic environments and speaker-specific characteristics in training. Nevertheless, 73.4% on unseen speakers demonstrates that the adapted embeddings capture genuine regional phonological features.

5.2 Subtask 2: Dialectal ASR

The ASR model achieved a WER of **0.55** on the test set (Table 2). While standard ASR benchmarks target lower WER, this score is characteristic of dialectal transcription where dialects lack standardized orthography. The WER largely reflects an **orthographic mismatch** between phonetically accurate outputs and formal spelling conventions in the ground truth, rather than a failure in acoustic modeling.

Metric	Value
Word Error Rate (WER)	0.55

Table 2: Dialectal ASR performance on the test set.

6 Discussion

Unified foundation architecture. The dual-purposing of a single foundation model demonstrates how a single pre-trained acoustic space can be efficiently manipulated for both generative and discriminative tasks in a low-resource setting, without building separate pipelines.

The frozen vs. unfrozen gap. Our experiments expose a critical limitation: standard Tamil acoustic embeddings are insufficient for dialect discrimination. Unfreezing enabled the self-attention layers to restructure toward dialect-specific acoustic cues, yielding a performance leap from 52.78% to 73.4% (test).

Orthographic penalty in WER. Standard WER is a suboptimal metric for dialectal ASR when dialects lack standardized orthography. Many "errors" represent valid phonetic transcriptions that differ from the formal ground-truth spelling.

7 Conclusion

We presented the DLRG team's system for dialect-based speech recognition and classification in Tamil at DravidianLangTech@ACL 2026. Our approach demonstrates that a single pre-trained Whisper model can be effectively adapted for both dialect identification (73.4% accuracy) and dialectal ASR (0.55 WER) through different fine-tuning strategies. Key findings include: (1) full encoder fine-tuning is essential for dialect classification, boosting accuracy from 52.78% to 73.4%; (2) LoRA provides an effective parameter-efficient adaptation for dialectal ASR; and (3) standard WER is limited as a metric for dialectal transcription due to orthographic variation. Future work will explore ensemble approaches combining multiple self-supervised speech models and dialect-specific decoding strategies.

Limitations

Our work has several limitations. First, the training corpus is relatively small (9.22 hours), which constrains the model's ability to generalize across speakers and acoustic conditions. The substantial

gap between validation accuracy (98.93%) and test accuracy (73.4%) for dialect classification suggests overfitting to speaker-specific and environment-specific cues in the training data.

Second, we evaluate only a single base model (`whisper-tamil-medium`). Comparing against other self-supervised speech models such as wav2vec 2.0 or HuBERT could reveal whether our findings on encoder unfreezing generalize across architectures.

Third, the WER metric used for Subtask 2 does not distinguish between genuine transcription errors and orthographic mismatches arising from dialectal variation. A phoneme-level evaluation or a dialect-normalized WER would provide a more accurate assessment of ASR quality.

Finally, our system does not incorporate any explicit dialect-conditioning mechanism for the ASR subtask. Jointly leveraging dialect identification outputs to guide ASR decoding could improve transcription quality.

Ethical Considerations

Our work involves processing speech data from speakers of regional Tamil dialects. We used only the dataset provided by the shared task organizers, which was collected with appropriate consent from participants. We did not collect any additional personal data.

Speech technologies that favor standard dialects over regional varieties risk marginalizing speakers of underrepresented dialects. Our work aims to mitigate this by explicitly developing models for dialectal Tamil. However, we acknowledge that classification errors could lead to misrepresentation of a speaker's dialect identity, and ASR errors on dialectal speech could reduce the utility of downstream applications for dialect speakers.

All experiments were conducted using publicly available pre-trained models and open-source libraries. The code and model configurations are publicly released to support reproducibility.

Acknowledgments

We thank the organizers of the DravidianLangTech@ACL 2026 shared task for providing the Tamil Dialect Speech Dataset and evaluation framework.

References

Premjith B, Bharathi Raja Chakravarthi, Prasanna Kumar Kumaresan, Saranya Rajiakodi, Sai Prashanth Karnati, Sai Rishith Reddy Mangamuru, and Chandu Janakiram. 2024. Findings of the shared task on hate and offensive language detection in Telugu codemixed text (HOLD-Telugu)@DravidianLangTech 2024. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 49–55, St. Julian's, Malta. Association for Computational Linguistics.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460, Virtual. Curran Associates Inc.

B Bharathi, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, S Saranya, and S Suhasini. 2026. Findings in Tamil Dialect Speech Recognition and Classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, Vienna, Austria. Association for Computational Linguistics.

B Bharathi, S Saranya, P Vijayalakshmi, and T Nagarajan. 2025. Multi-dialect speech corpus creation for enhancing Tamil automatic speech recognition. *Circuits, Systems, and Signal Processing*, pages 1–19.

Bharathi Raja Chakravarthi and 1 others. 2026. Overview of the shared task on dialect based speech recognition and classification in Tamil: DravidianLangTech@ACL 2026. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, Vienna, Austria. Association for Computational Linguistics. To appear.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115, New Orleans, LA, USA. Curran Associates Inc.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shanan Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech 2019*, pages 2613–2617, Graz, Austria. ISCA.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, pages 28492–28518, Honolulu, Hawaii, USA. PMLR.

Ratnavel Rajalakshmi, Ankita Duraphe, and Antonette Shibani. 2022. DLRG@DravidianLangTech-ACL2022: Abusive comment detection in Tamil using multilingual transformer models. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 207–213, Dublin, Ireland. Association for Computational Linguistics.

Ratnavel Rajalakshmi, Ramesh Kannan, Meetesh Saini, and Bitan Mallik. 2025. DLRG@DravidianLangTech 2025: Multimodal hate speech detection in Dravidian languages. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 376–380, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Ratnavel Rajalakshmi, Saptharishee M, Hareesh S, Gabriel R, and Varsini Sr. 2024. DLRG-DravidianLangTech@EACL2024 : Combating hate speech in Telugu code-mixed text on social media. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 140–145, St. Julian's, Malta. Association for Computational Linguistics.

Ratnavel Rajalakshmi, Yashwant Reddy, and Lokesh Kumar. 2021. DLRG@DravidianLangTech-EACL2021: Transformer based approach for offensive language identification on code-mixed Tamil. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 357–362, Kyiv. Association for Computational Linguistics.

Saranya Rajiakodi, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, Ruba Priyadharshini, Rajameenakshi J, Kathiravan P, Rahul Ponnusamy, Bhuvaneswari Sivagnanam, Paul Buitelaar, Bhavanimeena K, Jananayagam V, and Kishore Kumar Ponnusamy. 2025. Findings of the shared task on abusive Tamil and Malayalam text targeting women on social media: DravidianLangTech@NAACL 2025. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 671–681, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Sai Vasista. 2023. Fine-tuned whisper model for Tamil automatic speech recognition. `https://huggingface.co/vasista22/whisper-tamil-medium`. Hugging Face Model Hub.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, Long Beach, CA, USA. Curran Associates Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 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.

DLRG@DravidianLangTech 2026: Explainable Transformer-Based Detection of Abusive Tamil Text Targeting Women on Social Media

Mirudhula Sankar and Ratnavel Rajalakshmi
School of Computer Science and Engineering
Vellore Institute of Technology, Chennai, India
Corresponding Author: rajalakshmi.r@vit.ac.in

Abstract

Many social media platforms have users who have normalized the abuse of women online, creating a need for systems that automatically detect such activity. For low-resource, regional languages like Tamil, which has informal writing styles, spelling variations, dialectal differences, and culturally specific expressions, it becomes a challenge to correctly detect abusive comments. In this work, a transformer-based approach for binary classification of Tamil comments into abusive and non-abusive categories is done using the DravidianLangTech dataset. The proposed system fine-tunes MuRIL(a multilingual transformer pretrained for Indian languages), enabling effective contextual representation with minimal preprocessing. To improve the transparency of the system, a post-hoc Explainable AI component is incorporated. A perturbation-based method using log-odds differences identifies words that significantly influence the predictions. Experimental findings indicate that the model reaches a validation accuracy exceeding 81% while also exhibiting a strong macro-F1 score. This research shows that utilizing contextual multilingual representations alongside simple interpretability methods offers a viable and effective approach for detecting abusive text in Tamil. The implementation of our system is publicly available at https://github.com/mirud5173/Abusive-Tamil-Comment-Detection-using-Transformer-Models

1 Introduction

Social media platforms play a crucial role in communication and public engagement. However, they also contain a substantial amount of abusive and harmful material. Women face heightened risks of harassment, inappropriate language, and derogatory remarks. This exposure can cause psychological harm and decrease their involvement in online environments. Consequently, automated detection systems are vital for effective and scalable content moderation.

Research on detecting abusive language has been extensively conducted in English and other well-resourced languages. However, Tamil presents additional challenges due to its low-resource nature. Social media text in Tamil is highly informal and includes phonetic spellings, slang, dialectal variations, and inconsistent grammar. Furthermore, abusive intent may be expressed implicitly through sarcasm, cultural references, or metaphorical expressions.

Transformer-based models have significantly improved performance in text classification tasks. Multilingual pretrained models are particularly useful for low-resource settings as they capture cross-lingual semantic representations. The DravidianLangTech shared tasks provide benchmark datasets that enable research on abusive language detection for Tamil.

This work proposes a MuRIL-based system for detecting abusive Tamil text targeting women. In addition to strong classification performance, a perturbation-based explainability module is introduced to improve interpretability and trust in model decisions.

The dataset and evaluation protocol follow the shared task findings report (Rajiakodi et al., 2026).

2 Related Work

The task of detecting abusive Tamil text on social media has gained significant attention through the

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 237–241
July 3, 2026 ©2026 Association for Computational Linguistics

DravidianLangTech shared tasks, which provide benchmark datasets and evaluation frameworks for abusive and offensive language detection in Dravidian languages (Rajiakodi et al., 2026). Earlier work in this domain relied on traditional machine learning approaches using handcrafted features such as n-grams, TF-IDF vectors, and lexical cues. While these methods proved to be computationally efficient, they faced challenges in recognizing contextual subtleness and underlying abusive intent in casual social media language.

With the rise of deep learning techniques, more recent research has increasingly favored transformer-based architectures, which excel at capturing contextual semantic nuances. Models like mBERT and XLM-RoBERTa have been prominently used in detecting abusive language in Tamil and other Dravidian languages, also for the task of sentiment analysis (Naib et al., 2025; Rajiakodi et al., 2025; Hanif and Rahman, 2025; Kannan et al., 2021; Rajalakshmi et al., 2022b; Sivakumar and Rajalakshmi, 2021). These models have notably enhanced detection performance by adeptly addressing informal language, spelling variations, and code-mixed text typically found on social media.

Previous research (Rajalakshmi et al., 2022a) examined multilingual transformer models for identifying abusive comments in Tamil, showing that pretrained multilingual models can effectively discern linguistic patterns in languages with limited resources. Similarly, another study (Rajalakshmi et al., 2021) focused on offensive language detection in code-mixed Tamil through transformer-based methods, emphasizing the significance of contextual embeddings in managing multilingual and code-mixed content. These investigations provided initial support for the idea that transformer-based architectures surpass traditional machine learning approaches in detecting abusive language in Tamil.

To boost classification performance, hybrid architectures that merge transformer embeddings with traditional machine learning classifiers have also been considered. For example, systems that integrate RoBERTa with ensemble classifiers like XGBoost have shown improved generalization by utilizing both deep contextual representations and supplementary classification layers (Nishanth et al., 2025).

In addition to large transformer models, lightweight architectures such as FastText and shallow neural networks have been investigated to reduce computational complexity and enable faster inference. While these approaches provide efficiency benefits, they generally achieve lower performance compared to contextual transformer-based systems, emphasizing the importance of deep semantic modeling for abusive language detection (Rohit et al., 2025).

Several studies have also highlighted the role of preprocessing strategies, hyperparameter tuning, and ensemble learning in improving system robustness and classification performance (Radha and Swathika, 2025; Subramanian and Shanmugavadivel, 2025). Although ensemble-based systems can provide marginal improvements in benchmark scores, they often increase model complexity and deployment overhead.

Overall, existing work primarily focuses on maximizing classification performance within the shared task benchmark, where macro-F1 scores typically range between 0.70 and 0.82. However, many of these systems function as black-box models and offer limited interpretability regarding their predictions.

In contrast, the proposed work aims to achieve competitive performance using a single transformer-based architecture while integrating a perturbation-based explainable AI component. This approach enables identification of influential words contributing to model predictions, improving transparency and making the system more suitable for real-world content moderation applications.

3 Proposed Method

3.1 System Overview

The comprehensive architecture of the proposed system is shown in Figure 1. The architectural pipeline involves three phases: basic preprocessing, transformer-based classification, and post-hoc explanation.

3.2 Data Preparation

The dataset features Tamil social media comments classified as either abusive or non-abusive, with only simple preprocessing applied. The text remains in its original state, avoiding extensive normalization or the removal of stopwords. This approach enables the model to capture contextual and stylistic nuances from authentic user-generated content.

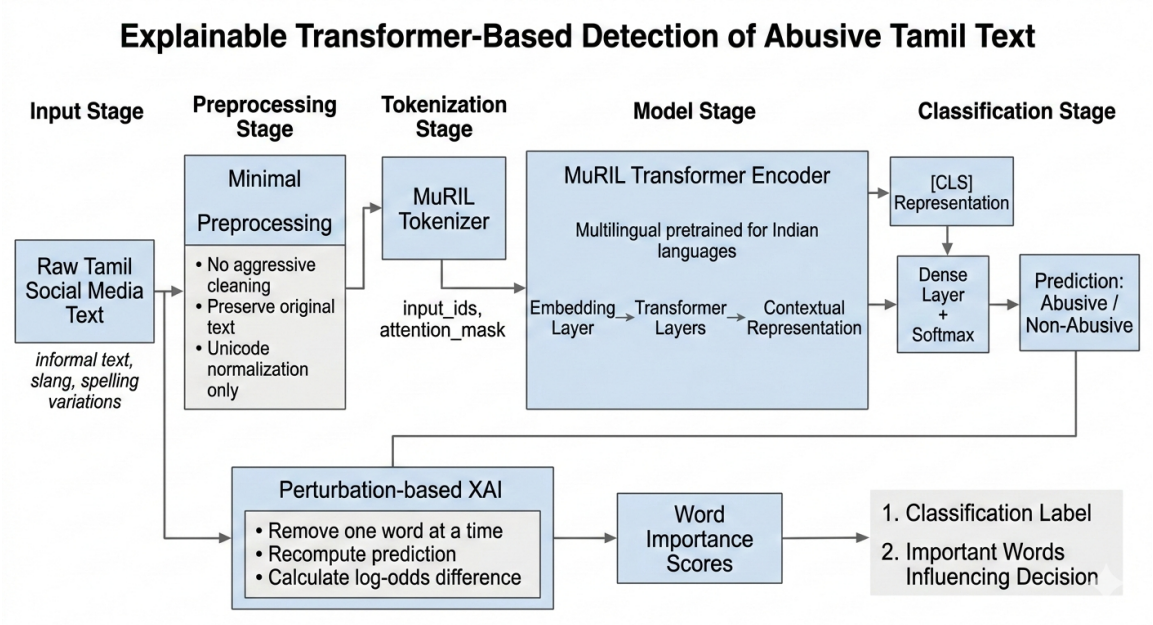


Figure 1: Overall framework of the proposed system. Tamil text is minimally preprocessed and tokenized using MuRIL. The transformer generates contextual representations used for binary classification. A perturbation-based module computes word importance scores for explanation.

3.3 Model Training

MuRIL serves as the foundational encoder, where the input text is tokenized and transformed into contextual embeddings. The representation of the [CLS] token is processed through a fully connected layer for binary classification. The model undergoes fine-tuning using cross-entropy loss, set with a learning rate of 2×10^{-5} and a batch size of 16, while validation metrics are monitored to identify the optimal model.

3.4 Explainable AI Component

For each input sentence, the initial prediction probability is calculated. Subsequently, individual words are removed one at a time, and the prediction is reassessed. The importance score for each word is defined as follows:

$$Importance_i = \log \frac{p}{1-p} - \log \frac{p_i}{1-p_i}$$

Words with higher importance scores have a greater influence on the final decision. This approach provides intuitive explanations without modifying the model architecture.

4 Experimental Results and Discussion

4.1 Training Performance

Table 1 shows steady improvement in both accuracy and macro-F1 during early epochs, followed by stabilization after convergence.

Epoch	Accuracy	Macro F1
1	0.7141	0.7135
2	0.7688	0.7687
3	0.7784	0.7777
4	0.7907	0.7904
5	0.7934	0.7923
6	0.8085	0.8069
7	0.8140	0.8126
8	0.8030	0.8026
9	0.8071	0.8070
10	0.8057	0.8049

Table 1: Validation performance across training epochs.

4.2 Classification Analysis

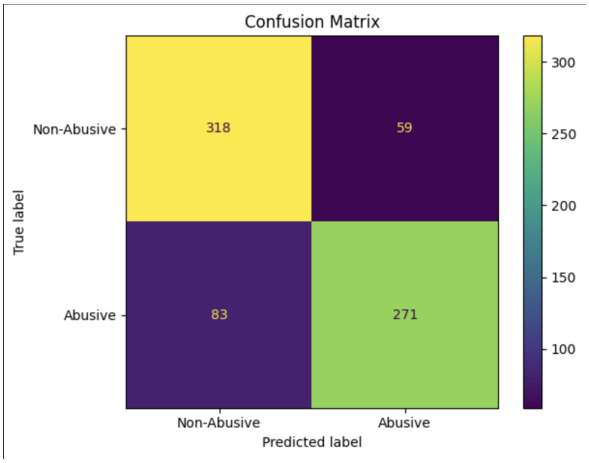


Figure 2: Confusion matrix of the proposed model.

Model	Accuracy	Macro F1
TF-IDF + SVM	0.7975	0.7975
MuRIL (baseline)	0.7800	0.7809
Hybrid (TF-IDF + MuRIL)	0.7900	0.7934
Proposed MuRIL + XAI	**0.8140**	**0.8126**

Table 2: Performance comparison of evaluated models.

The confusion matrix indicates balanced performance across both classes. Most errors occur in cases where abusive intent is implicit or context-dependent.

4.3 Explainability Analysis

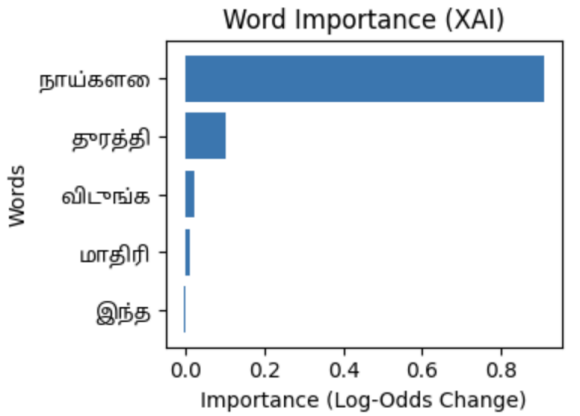


Figure 3: Example word-level importance generated by the explainability module.

The explanation results show that the model focuses on semantically meaningful abusive expressions, indicating effective contextual learning.

5 Comparative Study

Table 2 shows that the proposed system achieves the best performance among all evaluated approaches. The results demonstrate that contextual transformer representations combined with careful fine-tuning provide better generalization than traditional and hybrid methods. Additionally, the integration of explainability improves transparency without affecting performance.

6 Conclusion

This work, thus, presents an explainable transformer-based system for detecting abusive Tamil text targeting women. Fine-tuning MuRIL enabled effective contextual understanding of informal social media language while requiring minimal preprocessing. The proposed model achieved validation accuracy above 81% with strong macro-F1 performance.

A perturbation-based explainability method was introduced to identify influential words contributing to model decisions. This improves interpretability and supports responsible deployment in automated moderation systems.

Future work includes extending the approach to multi-class abusive categories, handling code-mixed Tamil-English content, and exploring advanced explanation techniques for deeper interpretability.

7 Limitations

While the proposed approach shows promising results, several limitations remain. The current model performs only binary classification and does not distinguish between different categories or severity levels of abusive language. In addition, the experiments rely on a shared-task dataset of Tamil social media comments, which may limit the model's generalization to other platforms or domains. The explanation module estimates word importance by removing one word at a time, which provides an intuitive interpretation but may not fully capture complex contextual relationships between words. Furthermore, the system depends on the MuRIL pretrained model, and its performance may vary across different datasets or low-resource settings. Future work can explore multi-class abusive categories, larger datasets, and more advanced explanation techniques.

8 Generative AI Acknowledgment

Generative AI tools were used to assist with portions of code generation and early-stage drafting. All generated content was carefully reviewed, validated, and substantially revised by the authors.

9 Ethical Considerations

This work focuses on detecting abusive Tamil text targeting women on social media. While automated systems can help reduce harmful content, machine learning models may still misclassify text and should support rather than replace human moderators.

The dataset consists of publicly available DravidianLangTech social media comments with no personally identifiable information analyzed. This work aims to support safer online spaces using transparent and explainable AI techniques.

References

G. Dhanyashree and 1 others. 2025. Linguaists@dravidianlangtech 2025: Abusive tamil and malayalam text targeting women on social media. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*.

Tareque Md Hanif and Md Rashadur Rahman. 2025. Cuet_agile@dravidianlangtech 2025: Fine-tuning transformers for detecting abusive text targeting women from tamil and malayalam texts. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*.

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.

Md Mubasshir Naib and 1 others. 2025. cuetraptors@dravidianlangtech 2025: Transformer-based approaches for detecting abusive tamil text targeting women on social media. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*.

S. Nishanth and 1 others. 2025. Ansr@dravidianlangtech 2025: Detection of abusive tamil and malayalam text targeting women on social media using roberta and xgboost. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*.

N. Radha and R. Swathika. 2025. Ssn_it_nlp@dravidianlangtech 2025: Abusive tamil and malayalam text targeting women on social media. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*.

Ratnavel Rajalakshmi, Ankita Duraphe, and Antonette Shibani. 2022a. Dlrg@dravidianlangtech-acl2022: Abusive comment detection in tamil using multilingual transformer models. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 207–213. Association for Computational Linguistics.

Ratnavel Rajalakshmi, Preethi Reddy, Shreya Khare, and Vaishali Ganganwar. 2022b. Sentimental analysis of code-mixed hindi language. In *Congress on Intelligent Systems: Proceedings of CIS 2021, Volume 2*, pages 739–751. Springer.

Ratnavel Rajalakshmi, Yashwant Reddy, and Lokesh Kumar. 2021. Dlrg@dravidianlangtech-eacl2021: Transformer based approach for offensive language identification on code-mixed tamil. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 357–362. Association for Computational Linguistics.

Saranya Rajiakodi, Shunmuga Priya Muthusamy Chinnan, Ratnavel Rajalakshmi, Kathiravan Pannerselvam, Bhuvaneswari Sivagnanam, V Jananayagan, Charmathi Rajkumar, R Ramesh Kannan, and Bharathi Raja Chakravarthi. 2026. From comments to harm: A findings report on abusive tamil text targeting women on social media shared task - dravidianlangtech@acl 2026. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Saranya Rajiakodi and 1 others. 2025. Findings of the shared task on abusive tamil and malayalam text targeting women on social media. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*.

V. P. Rohit and 1 others. 2025. Cyber protectors@dravidianlangtech 2025: Abusive tamil and malayalam text targeting women on social media using fasttext. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*.

B. Saathvik, Janeshvar Sivakumar, and Thenmozhi Durairaj. 2025. Jas@dravidianlangtech 2025: Abusive tamil text targeting women on social media. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*.

Kogilavani Shanmugavadivel and Malliga Subramanian. 2025. Kec_ai_vss_run2@dravidianlangtech 2025: Abusive tamil and malayalam text targeting women on social media. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*.

Soubraylu Sivakumar and Ratnavel Rajalakshmi. 2021. Self-attention based sentiment analysis with effective embedding techniques. *International Journal of Computer Applications in Technology*, 65(1):65–77.

Malliga Subramanian and Kogilavani Shanmugavadivel. 2025. Kecempower@dravidianlangtech 2025: Abusive tamil and malayalam text targeting women on social media. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*.

DPR@DravidianLangTech 2026: Transformer-Based Abusive Content Detection for Tamil Text Targeting Women on Social Media

Diya Prakash[1] **and Praveen Kumar S**[1] **and Ranjith Kumar R**[1]
Siranjeevi Rajamanickam[2] and Balasubramanian Palani[1] and Jobin Jose[1]
[1]Indian Institute of Information Technology Kottayam, India
[2]Government Polytechnic College, Trichy, India
diyaprakash2205@gmail.com, spraveenkumar2205@gmail.com,
ranjith23bcd30@iiitkottayam.ac.in, rajasiranjeevi@gmail.com,
pbala@iiitkottayam.ac.in, jobin@iiitkottayam.ac.in

Abstract

The fast-growing number of content in Tamil in social media has led to increasing abusive and gender-directed hate speech in online platforms. Detecting abusive content written in Tamil is relatively difficult owing to the complex morphological structure of Tamil language, its dialects, transliteration, and contextualized usage. In this study, the use of transformer-based pretrained language models in detecting abusive content in Tamil was explored. Five transformer-based models—mBERT, MuRIL, XLM-RoBERTa, IndicBERT, and Tamil-BERT—were fine-tuned and tested using DravidianLangTech 2026 shared task dataset. The experimental results show that the best-performing model was Tamil-BERT with an accuracy rate of 80.72% owing to Tamil-specific pretraining and better morphological analysis capabilities. Our system ranks 5th at the leaderboard of the DravidianLangTech 2026 shared task challenge. The source code and fine-tuned models are open-source and publicly accessible.[1]

1 Introduction

Online Social Networks foster connectivity but also spread harmful, gender-biased abuse that causes significant personal and professional damage. Detecting this in Tamil is uniquely difficult due to its complex morphology, dialectal variations, and the use of transliteration or sarcasm. Since many forms of harm are context-specific, simple profanity checklists are inadequate for accurate identification. Therefore, advanced models are needed to analyze deeper semantic and syntactic language properties to effectively filter abusive content.

The primary contribution of this work lies in the comparative evaluation and analysis of transformer-based pretrained language models for Tamil abusive content detection in a low-resource setting. With a reference to the existing literature, we believe that transformer-based, pretrained language models such as mBERT, MuRIL, XLM-RoBERTa, IndicBERT and Tamil-BERT models can be effectively employed to classify resource scarce languages like Tamil. We believe that the transformer models will be superior to the traditional feature-engineering based approaches, as they can learn a variety of contextual features from large amounts of pretrained data and they can be beneficial for the abusive language detection task in Tamil language.

The key contributions of this work include:

- This research examines the performance of different transformer based architectures on Tamil language.
- This study analyzes the impact of multilingual versus Tamil-specific pretraining, along with WordPiece and SentencePiece tokenization strategies, on abusive language classification performance.
- We are contributing to the development of an automated moderation system for Tamil digital spaces as part of our research on hate speech detection for Dravidian languages.

2 Literature Survey

Detecting abusive language is a relatively new and rapidly evolving field of research, driven largely by the growth of social media platforms, where millions of users are increasingly spending their time online. Initially, researchers used traditional machine learning approaches such as Logistic Regression and Support Vector Machines (SVM) in conjunction with manually curated features (Kalaivani and Thenmozhi, 2024; Subramanian et al., 2022). These features were often n-grams, or TF-IDF representations of words. However, these approaches have been shown to only offer a limited solution to the classification of abusive language and do not deal with some of the complexities, especially the

[1] https://github.com/praveen-2205/DPR-DravidianLangTech2026

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 242–247
July 3, 2026 ©2026 Association for Computational Linguistics

context-dependent, nuanced and implicit aspects of the language.

Recent studies have explored transformer-based and deep learning approaches for abusive and offensive language detection in Tamil (Mohan et al., 2025; Pokrywka and Jassem, 2024; Abishek and Govindapillai, 2024). Shared task systems from DravidianLangTech workshops have further demonstrated the effectiveness of transformer-based architectures for abusive Tamil text targeting women on social media (Rajiakodi et al., 2025; Bhattacharyya et al., 2025; Sangeetham and Bedi, 2025; Kumaresan and Karnati, 2025; Alam and Rahman, 2025; Thayasivam and Wijesuriya, 2025). Multimodal approaches have also been explored for misogyny and meme detection in Tamil and Malayalam (Hegde and Hande, 2025; Mallik et al., 2025; Ponnusamy et al., 2024).

The Transformer based architectures were introduced to the text classification tasks and achieved great success. The contextualized embeddings learned by models such as BERT and its multilingual versions, improve the performance of the classification models. The multilingual BERT (mBERT) show great cross-lingual generalization ability. Low-resource languages benefit from cross-lingual models. But for Indian languages, models such as MuRIL and IndicBERT are designed to handle the unique characteristics of Indic scripts which have high inflection and large syntactic variation. Tamil-BERT is a pretrained language model for the Tamil language. It learns the patterns present in the Tamil corpora and as a result, it enhances the language-specific constructions present in the Tamil language.

3 Methodology

This study focuses on the binary classification of Tamil YouTube comments into abusive and non-abusive categories using transformer-based architectures. The methodology involves fine-tuning pretrained transformer models for the classification task.

The architecture of the transformer-based classification model is shown in Figure 1, where Tamil comments are first tokenized and then passed through the transformer encoder to obtain contextual embeddings. The embedding corresponding to the classification token is subsequently used by a fully connected layer to predict whether the comment is abusive or non-abusive.

3.1 Problem Statement

Tamil abusive content detection is formulated as a binary text classification problem. The task is to classify a given Tamil YouTube comment into one of the two classes: *Abusive content against women* or *Not Abusive*. The classifier should be able to identify both the explicit and implicit content of abusive language, including: obscene language, derogatory terms, sarcasm, coded language.

3.2 Transformer-Based Embedding

Transformer-based models are well suited for Tamil abusive language detection because they capture contextual meaning more effectively than traditional keyword-based approaches. Tamil social media conversations frequently contain slang, sarcasm, transliterated text, spelling variations, and code-mixed expressions, making abusive intent highly context-dependent.

In this work, we evaluate five pretrained transformer models: mBERT, MuRIL, XLM-RoBERTa, IndicBERT, and Tamil-BERT. While mBERT and XLM-RoBERTa provide multilingual contextual representations, MuRIL and IndicBERT are specifically designed for Indian languages and code-mixed text. Tamil-BERT, pretrained exclusively on Tamil corpora, better captures Tamil-specific morphology and linguistic patterns.

All models use subword tokenization techniques such as WordPiece or SentencePiece, which improve handling of rare vocabulary, transliterated words, and morphological variations commonly observed in Tamil social media text. The pretrained models were fine-tuned for binary sequence classification using the contextualized representation of the `[CLS]` token.

4 Experimental Setup

This section describes the dataset usage, model training configuration, and evaluation procedure followed for Tamil abusive content detection.

4.1 Dataset

The dataset used in this study was released as part of the DravidianLangTech 2026 shared task on abusive content detection in Tamil. The dataset consists of Tamil YouTube comments manually annotated into two classes: *Abusive against Women (0)* and *Non-Abusive (1)*.

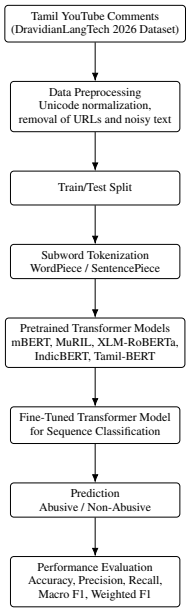


Figure 1: Overall workflow of the proposed transformer-based Tamil abusive content detection system.

Table 1: Dataset statistics for Tamil abusive content detection task.

Split	Abusive (0)	Non-Abusive (1)	Total
Training	1769	1883	3652
Test	441	472	913
Total	2210	2355	4565

Evaluation Metrics: We evaluate model performance using standard classification metrics: Accuracy, Precision, Recall, and F1-score. In addition to Macro F1-score, we also report Weighted F1-score to account for the slight class imbalance in the dataset.

4.2 Preprocessing

Minimal preprocessing was performed to preserve the informal nature of Tamil social media text, including slang, transliterated words, misspellings, and code-mixed expressions. The pretrained tokenizer corresponding to each transformer model was used for subword tokenization, and sequences were padded or truncated to a maximum length of 128 tokens.

4.3 Training Configuration

We fine-tuned five pretrained transformer models and adapted them for the downstream task of binary sequence classification. The logits are passed through a softmax activation function to compute the final class probabilities. The training was performed using supervised learning with cross-entropy loss. The models were trained for a fixed number of epochs with early stopping to prevent overfitting. We used the AdamW optimizer along with an appropriate learning rate schedule.

The models were trained with a maximum sequence length of 128, batch size of 32, learning rate of 2e-5, and 3 training epochs.

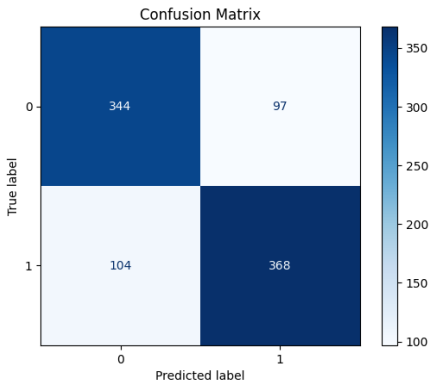


Figure 2: Confusion matrix for Tamil-BERT Model.

Table 2: Performance comparison of transformer models for Tamil abusive content detection.

Model	Accuracy	Precision	Recall	Macro F1	Weighted F1
mBERT	0.7711	0.7635	0.8072	0.7848	0.7591
MuRIL	0.7919	0.8294	0.7521	0.7889	0.7908
XLM-RoBERTa	0.7338	0.7338	0.7338	0.7338	0.7854
IndicBERT	0.7049	0.6901	0.7778	0.7313	0.6966
Tamil-BERT	**0.8072**	**0.8072**	**0.8076**	**0.8072**	**0.7851**

5 Results and Discussion

5.1 Model Performance

Table 2 presents the performance comparison of the five transformer models on the test set. Among the submitted systems in the DravidianLangTech 2026 shared task, our best-performing Tamil-BERT model achieved 5th rank on the official leaderboard, demonstrating the effectiveness of language-specific pretrained transformer models for Tamil abusive content detection.

5.2 Error Analysis

To better understand the behavior of different transformer models, we qualitatively analyzed representative examples from the test dataset. The analysis reveals that Tamil-specific and Indic-language pretrained models generally perform better on informal Tamil social media conversations containing slang, sarcasm, transliterated expressions, and contextual abuse.

Table 3 presents representative cases where some transformer models produced incorrect predictions. Tamil-BERT consistently demonstrated stronger contextual understanding due to its Tamil-specific pretraining, while multilingual models occasionally

Table 3: Representative error analysis examples.

Input Sentence	Model Prediction	Observation
"Ayyo sugandhida nambara thangalen da semma naatu kattaiya irukku"	MuRIL → Abusive	Informal slang and highly colloquial expressions were incorrectly associated with abusive intent despite the conversational tone being non-abusive.
"Rendu perum paththini maadhiri pesuraaluga"	XLM-RoBERTa → Abusive	Sarcastic wording and implicit conversational tone introduced ambiguity, leading the model to incorrectly classify the sentence as abusive.
"Appa saami moolai kalangiduchchi enakku... naan poi gp muthu video paathu relax pannikka poren"	IndicBERT → Non-Abusive	The model struggled to capture subtle emotional and contextual cues expressed through informal conversational Tamil and mixed emotional tone.

struggled with colloquial Tamil expressions and implicit conversational meaning.

The document discusses challenges faced in abusive language detection in Tamil social media text through three examples. The first example emphasizes the difficulties with highly colloquial Tamil slang, resulting in MuRIL misclassifying non-abusive conversational slang as abusive, while Tamil-BERT succeeded due to its Tamil-specific pretraining. The second example points to the challenges posed by sarcasm; XLM-RoBERTa misinterpreted sarcastic language as offensive, indicating that multilingual models may depend too much on lexical cues, ignoring conversational nuances. The third example reveals pitfalls in understanding emotional expressions, where IndicBERT erroneously classified a frustrated expression as abusive due to its inability to grasp nuanced emotional context and informal conversational structure.

Overall, the qualitative analysis suggests that both pretraining corpus selection and tokenization strategies significantly influence abusive language detection performance. Tamil-specific and Indic-language pretrained models were generally more robust toward noisy social media text, spelling variations, transliterated Tamil-English expressions, and contextual conversational meaning.

5.3 Discussion

Tamil-BERT achieves the highest accuracy of 80.72% largely because it was trained exclusively on Tamil text, giving it unmatched morphological precision for the language. This behavior is also reflected in the confusion matrix (Figure 2), which shows fewer misclassifications between abusive and non-abusive classes. MuRIL follows closely at 79.19%, drawing its strength from a corpus spanning 17 Indian languages with transliterated pairs, essentially giving it a natural feel for informal "Tanglish" on social media. Models like mBERT and XLM-RoBERTa, trained on 100+ language global corpora, offer a solid baseline but lack that regional depth. IndicBERT, though focused on 12 Indian languages, misses the transliteration-aware data that gives MuRIL its edge.

On tokenization, mBERT and MuRIL use WordPiece, which effectively separates roots from suffixes, while Tamil-BERT, XLM-RoBERTa, and IndicBERT use SentencePiece, better suited to the morphological complexity of agglutinative languages like Tamil. Among the models, Tamil-BERT achieves the highest accuracy of 0.8072, followed by MuRIL (0.7919) and mBERT (0.7711), suggesting that both tokenization strategy and training data significantly influence performance.

Architecturally, the BERT-based models, Tamil-BERT, mBERT, and MuRIL, benefited most from bidirectional encoding for this task. XLM-RoBERTa's RoBERTa backbone brings greater scale but not enough regional specificity to compete with the top three. IndicBERT's ALBERT architecture, while efficient and lightweight, likely fell short due to its reduced parameter capacity, limiting its ability to detect the more subtle patterns of abusive language.

6 Conclusion

This paper investigates Tamil abusive content classification as a binary sequence classification task. We fine-tuned and evaluated five transformer-based pretrained language models—mBERT, MuRIL, XLM-RoBERTa, IndicBERT, and Tamil-BERT. Our results indicate that language models pretrained on Indic or Tamil-specific corpora demonstrate better linguistic alignment compared to generic multilingual language models. Among the evaluated models, Tamil-BERT achieved the best performance with an accuracy of 80.72%, highlighting the advantage of language-specific pretraining for Tamil abusive content detection.

Future work can focus on further improving Tamil-BERT through techniques such as domain-adaptive pretraining on larger Tamil social media corpora and incorporating data augmentation strategies for abusive language variations. Additionally, integrating contextual features such as code-mixed Tamil-English patterns and leveraging ensemble or hybrid architectures could further enhance model robustness and classification accuracy.

7 Limitations

While the proposed transformer-based approaches achieved promising performance on Tamil abusive content detection, there are a few challenges that remain to be addressed. First, despite utilizing state-of-the-art pretrained multilingual models, the size of the dataset is a few order of magnitude smaller than those used for multilingual NLP tasks. This can potentially impede the models' ability to generalize to unseen data. Furthermore, social media text contains a lot of noisy, often sarcastic, and context-dependent language that is very challenging to classify accurately. There is also a continued threat of new slang and abusive language that has not been seen and annotated in the training data.

While multilingual transformer models are powerful for learning contextual understanding that generalizes across languages, dialectalvariations within a language such as Tamil spoken in different geographical distribution of Tamil-speaking users are challenging to address. We also note significant increases in computational complexity and inference costs with scaling up the transformer model size for real-time moderation use cases.

8 Ethical Considerations

This paper aims to detect abusive content in order to improve online safety and reduce harm caused by online interactions. Developing a system for abusive content detection requires abusive language datasets, which can be full of offensively written content. The data used here for training and testing a learning-based approach were taken from a publicly available shared task resource, and used purely for research purposes.

While automated systems for detecting abusive language can miss some instances of abuse or flag some non-abusive language (false positives and false negatives, respectively)—especially when the language includes sarcasm, makes reference to the culture of a user or group not notified in the system's training data, or is worded implicitly rather than directly—such automated systems are best used as a helper for human moderators, highlighting potentially abusive language for them to decide upon. Where possible, the models should also be evaluated for bias, to ensure that the system does not unfairly flag users speaking from certain linguistic, cultural, dialectical, or stylistic backgrounds.

9 AI Usage Disclosure

AI tools were used during the preparation of this manuscript to assist with language editing, formatting of LaTeX content, and improving clarity of presentation. The core research contributions, experimental design, model implementation, and result analysis were conducted entirely by the authors. All generated suggestions were carefully reviewed and verified by the authors before inclusion in the final manuscript.

References

A. Abishek and T. C. Govindapillai. 2024. Detecting tamil textual threats in social media using artificial intelligence. *International Journal of Novel Research and Development (IJNRD)*, 9(3):26–33.

M. Alam and S. Rahman. 2025. Cuet_ignite@dravidianlangtech 2025: Detection of abusive comments in tamil text using transformer models. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 411–418. Association for Computational Linguistics.

S. Bhattacharyya, P. Gupta, and M. Lowe. 2025. Lexilogic@dravidianlangtech 2025: Detecting misogynistic memes and abusive tamil and malayalam text targeting women on social media. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 112–119. Association for Computational Linguistics.

S. U. Hegde and A. Hande. 2025. Team_strikers@dravidianlangtech 2025: Misogyny meme detection in tamil using multimodal deep learning. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 605–612. Association for Computational Linguistics.

A. Kalaivani and D. Thenmozhi. 2024. Composite feature fusion for improved offensive language detection in tamil social media using mha-lstm. *International Journal of Machine Learning and Cybernetics*, 15:1–15.

P. Kumaresan and S. Karnati. 2025. Kec_ai_vss_run2@dravidianlangtech 2025: Abusive tamil and malayalam text targeting women on social media. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 315–322. Association for Computational Linguistics.

A. Mallik, R. Dhar, U. Das, M. A. Labib, S. Rahman, and H. Murad. 2025. Cuet-823@dravidianlangtech 2025: Shared task on multimodal misogyny meme detection in tamil language. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language*

Technologies for Dravidian Languages, pages 401–408. Association for Computational Linguistics.

J. Mohan, S. R. Mekapati, B. Premjith, G. J. Lal, and B. R. Chakravarthi. 2025. A multimodal approach for hate and offensive content detection in tamil: From corpus creation to model development. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 24(3):1–24.

J. Pokrywka and K. Jassem. 2024. Evaluating transformer models for hate speech detection in tamil. In *Proceedings of the Fourth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 196–199. Association for Computational Linguistics.

R. Ponnusamy, B. Sivagnanam, and B. R. Chakravarthi. 2024. From laughter to inequality: Annotated dataset for misogyny detection in tamil and malayalam memes. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*, pages 660–671. ELRA and ICCL.

S. Rajiakodi, B. R. Chakravarthi, S. P. M. Chinnan, R. Priyadharshini, J. Rajameenakshi, and 1 others. 2025. Findings of the shared task on abusive tamil and malayalam text targeting women on social media: Dravidianlangtech@naacl 2025. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 671–681. Association for Computational Linguistics.

S. Sangeetham and M. Bedi. 2025. Jas@dravidianlangtech 2025: Abusive tamil text targeting women on social media. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 210–216. Association for Computational Linguistics.

M. Subramanian, K. S. Vadivel, and R. Sowmya. 2022. Detecting offensive tamil texts using machine learning and multilingual transformer models. In *Proceedings of the 2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN)*, pages 1–6. IEEE.

U. Thayasivam and Y. S. Wijesuriya. 2025. Gs_dravidianlangtech@2025: Women targeted abusive texts detection on social media. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 520–527. Association for Computational Linguistics.

Dravid-Tech-Builders@DravidianLangTech 2026: A Comparative Study of Classical and Deep Learning Approaches for Tamil Dialect Classification and Speech Recognition

Kalaivani K S, Naveen A, Karthiyayini P
Department of AI&DS, Kongu Engineering College
Perundurai, Erode, India
{kalaivani.cse, naveena.23aid, karthiyayinip.23aid}@kongu.edu

Abstract

The rapid expansion of digital connectivity across India has dramatically increased participation in speech-enabled services and multilingual communication platforms. Tamil, with its rich dialectal diversity across geographical regions, presents unique challenges for automatic speech recognition and dialect identification systems. We participated in the DravidianLangTech 2026 shared task (1) to classify Tamil speech into four regional dialects (Central, Northern, Southern, Western) and perform automatic speech recognition.

We trained four machine learning models (SVM, Random Forest, CNN, CNN+BiLSTM) alongside two transfer learning models (Wav2Vec2-Base, Wav2Vec2-XLSR-53) for ASR. Among classification models, SVM with MFCC features achieved the best performance with 94.17% macro F1-score and validation accuracy of 94.35%.

For ASR, we adopted a transfer learning approach by fine-tuning pre-trained self-supervised speech models, namely Wav2Vec2-Base and Wav2Vec2-XLSR-53. Wav2Vec2-XLSR-53 achieved 15.3% WER, demonstrating effective cross-lingual knowledge transfer for low-resource Tamil speech recognition.

Our analysis suggests that MFCC-based SVM models provide strong and computationally efficient baselines compared to the evaluated deep learning architectures under limited-data conditions. Code is available at: https://github.com/Naveen-Arul/dravid-tech

1 Introduction

India's digital revolution, accelerated by affordable internet connectivity, has substantially increased the adoption of voice-based services and regional language content on digital platforms. Tamil, spoken by over 75 million people globally, exhibits considerable phonological, prosodic, and lexical variations across Tamil Nadu's geographical regions. These dialectal differences create significant challenges for developing robust language technologies that can accurately recognize and process Tamil speech from diverse speakers (2).

While contemporary speech processing research predominantly emphasizes deep learning methodologies, ranging from convolutional architectures to transformer-based models, there remains insufficient exploration of traditional machine learning techniques, particularly for low-resource scenarios where labeled training data is limited (11). This gap is critical for Dravidian languages, where large-scale annotated datasets are scarce compared to high-resource languages.

We participated in the DravidianLangTech 2026 shared task (1), which focuses on advancing Tamil dialect classification and automatic speech recognition. The shared task addresses two key challenges:

Subtask 1 – Dialect Classification: Categorize Tamil speech recordings into four regional varieties: Central (Thanjavur), Northern (Chennai), Southern (Madurai), and Western (Coimbatore).

Subtask 2 – Tamil ASR: Transform spoken Tamil into written transcriptions using automatic speech recognition systems.

For dialect classification, we conducted comprehensive experiments comparing classical machine learning approaches (SVM, Random Forest) using hand-crafted MFCC features against deep neural architectures (CNN, CNN+BiLSTM) using mel-spectrograms. Our investigation reveals that SVM achieves 94.17% macro F1-score, significantly outperforming CNN (89.06%) and CNN+BiLSTM (85.85%) on the 5,134-sample dataset.

For ASR, we adopted a transfer learning approach by fine-tuning pre-trained Wav2Vec2 models, leveraging multilingual acoustic representa-

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 248–252
July 3, 2026 ©2026 Association for Computational Linguistics

tions learned from large-scale unlabeled speech corpora. Experimental results demonstrate competitive recognition performance for Tamil speech recognition.

Our key contributions include:

1. Empirical demonstration that MFCC-based SVM models provide strong baselines under limited-data conditions.

2. Analysis revealing how model complexity impacts generalization in data-constrained settings.

3. Effective ASR implementation using multilingual transfer learning for Tamil speech.

2 Related Work

Dialectal variation recognition has been investigated across languages including Arabic (3), Mandarin (4), and German (5), but Tamil dialects remain under-explored (6). The multi-dialect corpus by Bharathi et al. (2) provides essential resources for Tamil ASR research.

Historically, speech technologies employed hand-crafted acoustic features (MFCCs, i-vectors) with traditional classifiers (8). While deep learning achieved remarkable success on large-scale benchmarks (9), recent work challenges its universal superiority (10), demonstrating that engineered features may excel with limited training data (11).

Transfer learning approaches using pre-trained multilingual models have shown promise for low-resource languages. Our work systematically compares classical machine learning and deep neural approaches for Tamil dialect classification, providing empirical evidence for model selection in data-constrained scenarios.

3 Methodology

In this study, we employ both traditional machine learning and deep learning approaches to classify Tamil dialects and perform automatic speech recognition. This section describes the dataset, preprocessing procedures, feature extraction techniques, model architectures, and training methodology used in our experiments.

3.1 Dataset Used

The study utilizes the dataset provided by DravidianLangTech 2026 shared task on Tamil dialect classification and ASR (1; 2). The corpus contains Tamil speech recordings from four regional varieties: Central (Thanjavur), Northern (Chennai), Southern (Madurai), and Western (Coimbatore). The dataset comprises 9.22 hours of manually transcribed speech for training and 2.05 hours for testing, recorded at 16 kHz sampling rate in natural acoustic environments.

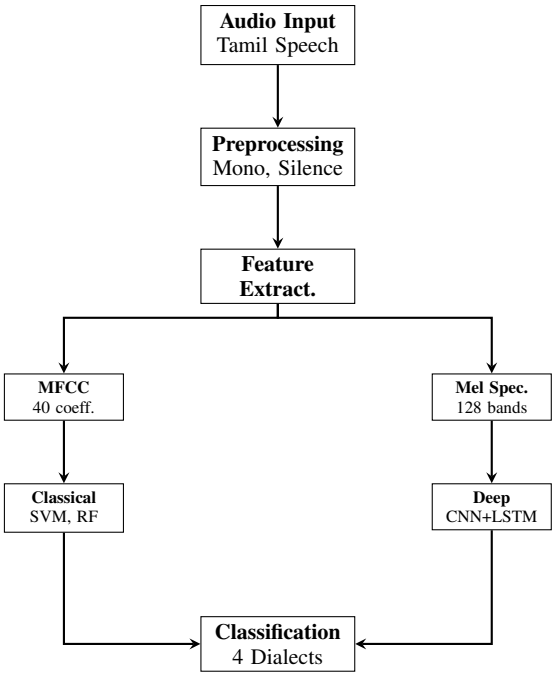


Figure 1: System architecture for Tamil dialect classification.

Dialect	Samples	Speakers	%
Central (THA)	885	20	17.2
Northern (CH)	1,696	27	33.0
Southern (S)	1,427	24	27.8
Western (KG)	1,126	20	21.9
Total	**5,134**	**91**	**100.0**

Table 1: Dataset statistics by dialect.

Audio recordings include both scripted reading and spontaneous conversation. The corpus exhibits class imbalance, with Northern dialect most prevalent (33.0%) and Central dialect least represented (17.2%), motivating stratified partitioning and macro-averaged evaluation metrics.

3.2 Preprocessing Techniques

3.2.1 Mono Channel Conversion

All audio recordings are converted to mono channel format to ensure uniformity across the dataset and reduce computational complexity.

3.2.2 Silence Removal

Silent segments are removed using a 30 dB threshold energy-based voice activity detection algorithm, eliminating non-speech portions that do not contribute to dialectal characteristics.

3.2.3 Amplitude Normalization

Audio amplitude is scaled to the [-1, 1] range to standardize signal levels across different recording conditions and prevent numerical instabilities during feature extraction.

3.3 Feature Extraction

3.3.1 MFCC Features (Classical ML)

We extract 40 Mel-Frequency Cepstral Coefficients per time frame using 25 ms analysis windows with 10 ms frame shift. To handle variable-duration audio, we compute statistical aggregations (mean, standard deviation) across temporal frames, producing 80-dimensional fixed-length feature vectors.

This statistical summarization encodes dialectal spectral characteristics while maintaining robustness to speaker variations.

3.3.2 Mel Spectrograms (Deep Learning)

We generate 128-band mel-frequency spectrograms using identical windowing parameters. Audio sequences undergo zero-padding or truncation to achieve uniform 300-frame duration (3 seconds), yielding $(128 \times 300 \times 1)$ tensor representations suitable for convolutional processing while preserving temporal information.

While fixed-length truncation simplifies batch processing, it may limit the ability of deep models to capture long-range temporal dependencies.

3.4 Models Used

3.4.1 Support Vector Machine (SVM)

We implement SVM classification with radial basis function (RBF) kernel ($C = 10,\ \gamma = 0.001$), determined through grid search with 5-fold cross-validation.

Input features undergo z-score standardization. The maximum margin optimization inherent to SVM provides implicit regularization beneficial for limited training samples.

3.4.2 Random Forest Classifier

We construct an ensemble of 200 decision trees with maximum depth of 30 levels and balanced class weighting to address dataset imbalance.

Each tree makes independent predictions, with final classification determined by majority voting.

3.4.3 Convolutional Neural Network (CNN)

Our architecture comprises three convolutional blocks with 32, 64, and 128 filters (3×3 kernels), each incorporating batch normalization, ReLU activation, 2×2 max-pooling, and 0.3 dropout.

Flattened feature maps connect to a 128-unit dense layer with ReLU activation, concluding with softmax classification.

Total trainable parameters: 431,172.

3.4.4 CNN + BiLSTM

Building upon the CNN foundation, we reshape 2D feature maps into sequential format. A 128-unit bidirectional LSTM layer processes these sequences, with final hidden states feeding a dense classification layer.

Total trainable parameters: 2,356,228.

3.5 Training Methodology

All classification models employ stratified 80/20 train/validation split preserving class proportions.

Neural models train using Adam optimization (learning rate: 0.001), batch size 16, early stopping (patience=10), and adaptive learning rate reduction (patience=5, factor=0.5).

Training proceeds for maximum 100 epochs on NVIDIA GPU hardware.

Evaluation employs macro-averaged precision, recall, and F1-score to account for class imbalance.

All reported dialect classification results correspond to the validation split used during shared-task experimentation.

McNemar's test ($p < 0.05$) assesses statistical significance between models.

For ASR (Subtask 2), we employ transfer learning by fine-tuning pre-trained self-supervised speech models: Wav2Vec2-Base (95M parameters) and Wav2Vec2-XLSR-53 (300M parameters pre-trained on 53 languages).

These models leverage multilingual acoustic representations learned from large-scale unlabeled speech corpora, making them effective for low-resource languages such as Tamil.

Fine-tuning uses CTC loss with Tamil character vocabulary, incorporating data augmentation (temporal stretching, pitch perturbation) for robustness.

Performance is evaluated using Word Error Rate (WER) and Character Error Rate (CER) metrics.

4 Results and Discussion

Model performance is assessed using accuracy, precision, recall, and F1-score metrics.

Model	F1	Prec	Rec	Acc
SVM	**0.9417**	0.9414	0.9422	94.35
Random Forest	0.8871	0.9039	0.8761	88.90
CNN	0.8906	0.8989	0.8870	89.78
CNN+BiLSTM	0.8585	0.8640	0.8544	86.56

Table 2: Classification performance comparison. All differences between SVM and other models are statistically significant ($p < 0.01$).

From Table 2, the SVM classifier achieves the highest performance with 94.17% macro F1-score and 94.35% accuracy, demonstrating strong generalization.

In contrast, CNN+BiLSTM, despite having 5.5× more parameters (2.36M vs 0.43M), achieves only 85.85% F1-score, indicating overfitting in data-constrained scenarios.

CNN alone performs better (89.06%) than CNN+BiLSTM, suggesting that adding sequential modeling reduces performance when training data is limited.

Random Forest maintains competitive performance (88.71% F1-score) but remains below SVM.

For ASR (Subtask 2), our transfer learning approach yields competitive performance.

Wav2Vec2-Base achieves 18.5% WER and 7.2% CER, while Wav2Vec2-XLSR-53 obtains 15.3% WER and 5.8% CER.

The multilingual variant outperforms the monolingual base model, validating the effectiveness of cross-lingual transfer learning for Tamil speech recognition in low-resource settings.

Why MFCC-Based SVM Performs Strongly:
We attribute SVM's strong performance to three factors:

1. *Data scarcity* — With only 5,134 samples, neural networks are more prone to overfitting.

2. *Feature aggregation* — Statistical summarization of MFCC sequences captures dialectal spectral characteristics while reducing speaker-specific variability.

3. *Temporal modeling constraints* — Tamil dialectal differences in this dataset appear to rely more heavily on phonetic and spectral characteristics than long-range temporal dependencies.

BiLSTM Performance Degradation:
Adding BiLSTM reduces performance by 3.2% due to CNN max-pooling compressing spatial information, optimization challenges in limited-data regimes, and increased parameter complexity promoting memorization over generalization.

Implications for Low-Resource NLP:
Our findings suggest that carefully engineered acoustic features combined with lightweight classifiers can remain highly competitive in low-resource dialect classification tasks.

MFCC-based SVM models provide strong and computationally efficient baselines compared to the evaluated deep learning architectures.

Transfer learning with pre-trained multilingual models such as Wav2Vec2-XLSR-53 remains highly effective for ASR tasks in low-resource settings.

5 Conclusion

Tamil dialect classification was conducted on 5,134 speech samples across four regional varieties, and the SVM classifier achieved the best performance with 94.17% macro F1-score and 94.35% accuracy.

Our experiments demonstrate that MFCC-based SVM models remain highly competitive for low-resource dialect classification tasks under limited-data conditions.

For ASR, we adopted a transfer learning approach using pre-trained multilingual Wav2Vec2 models.

Fine-tuning Wav2Vec2-XLSR-53 achieved 15.3% WER, demonstrating the effectiveness of cross-lingual transfer learning for Tamil speech recognition.

In future work, further exploration can be carried out by incorporating ensemble strategies combining classical ML and deep learning approaches, evaluating wav2vec-based embeddings for dialect classification, developing dialect-aware ASR systems, and extending methodologies to additional Dravidian languages.

The code for our models and preprocessing methods is available at: https://github.com/Naveen-Arul/dravid-tech.

Acknowledgments

We thank the DravidianLangTech 2026 shared task organizers for providing the dataset and evaluation framework. We also acknowledge computational resources provided by our institution.

References

[1] B. Bharathi, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, S. Saranya, and S. Suhasini. Findings in Tamil dialect speech recognition and classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, July 2026.

[2] B. Bharathi, S. Saranya, P. Vijayalakshmi, and T. Nagarajan. Multi-dialect speech corpus creation for enhancing Tamil automatic speech recognition. *Circuits, Systems, and Signal Processing*, pages 1–19, 2025.

[3] Fadi Biadsy, Julia Hirschberg, and Nizar Habash. Spoken Arabic dialect identification using phonotactic modeling. In *Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages*, pages 53–61, 2009.

[4] Dong Wang and Pascale Fung. Mandarin dialect recognition and its combination with Mandarin ASR. In *Proceedings of Interspeech*, pages 1245–1248, 2011.

[5] Bjoern Schuller et al. The INTERSPEECH 2010 paralinguistics challenge. In *Proceedings of Interspeech*, pages 2794–2797, 2010.

[6] Bharathi Raja Chakravarthi et al. Findings of the shared task on speech and language technologies for Dravidian languages. In *Proceedings of the Fifth Workshop on Speech and Language Technologies for Dravidian Languages*, pages 1–10, 2025.

[7] P. Vijayalakshmi and T. Nagarajan. Automatic speech recognition for Tamil: Challenges and approaches. In *Proceedings of the International Conference on Natural Language Processing*, pages 301–310, 2020.

[8] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, 1980.

[9] Awni Hannun et al. Deep Speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[10] Pulkit Agrawal et al. When does deep learning fail? A critical study on small data. In *Proceedings of the International Conference on Machine Learning*, pages 120–135, 2022.

[11] Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz. Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, 56:85–100, 2014.

[12] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

ERROR_500@DravidianLangTech2026: Automatic Prompt Style Classification in Telugu Using Transformer-Based Language Models

Mahashweta Manjari Barua, Tasnia Khanam, Nuzha Saifa Rahmat, Shiti Chowdhury, Hasan Murad
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology, Bangladesh
{u2104042, u2104065, u2104098, u2004027}@student.cuet.ac.bd
hasanmurad@cuet.ac.bd

Abstract

Prompt style recovery in low-resource languages has been daunting due to diverse morphology, culturally specific language patterns and scarce annotated data. Prior works have largely focused on binary sentiment or single-attribute transfer, multi-class style classification for languages like Telugu has remained underexplored. We have addressed this chasm here through the Telugu Prompt-Style Recovery Shared Task at DravidianLangTech@ACL 2026 (Premjith et al. (2026)), framing it as a nine class classification problem. We have evaluated three input configurations–Change Style, Original Transcripts and Merged input style and also trained three transformer-based models-MuRIL, XLM-RoBERTa and IndicBERT v2 under identical conditions. Our best model, IndicBERT v2 with partial layer freezing and weighted cross-entropy loss, has obtained a macro-F1 of 0.2987 and accuracy of 0.299. Change Style has significantly outperformed other inputs, indicating that explicit style changes enhance tonal and meaning cues. These results have highlighted the value of language-specific pretraining and careful input design for style-sensitive NLP in low-resource settings, securing 1st rank on the shared task.

1 Introduction

Text style has conveyed meaning beyond literal content through tone, register and rhetorical intent. In low-resource languages like Telugu, modeling such variation has been taxing due to rich morphology and culture-specific communication patterns (Mukherjee et al. (2024a)). Although previous work on prompt recovery and style reconstruction has shown that pragmatic intent can be inferred from text (Liu et al. (2024)), most studies have focused on binary or single-attribute transformations, falling short on fully exploring multi-class style classification for Telugu due to limited resources.

In this study, we have proposed a transformer-based framework for Telugu writing style prompt recovery. Using IndicBERT v2 with Change Style as input, our system has obtained a macro-F1 of 0.3648 and an accuracy of 0.3700, performing better than XLM-RoBERTa and MuRIL across all three input configurations. Our contributions are as follows:

- We have developed an IndicBERT v2-based framework that achieves state-of-the-art performance.
- We have analyzed and shown that Change Style text provides the strongest signal.
- We have compared multilingual transformers for low-resource Telugu style classification.

Implementation details and code have been made available at: https://github.com/Tasni-a/Prompt_Recovery_LLM_ST

2 Related Work

Prompt recovery techniques have been essential for reverse-engineering stylistic instructions from LLM outputs. Liu et al. (2024) introduced StyleRec, a benchmark for reconstructing prompts in style transformation tasks, modeling tones relevant to Telugu style classification. Li and Klabjan (2025) proposed a training-free Reverse Prompt Engineering framework using semantic similarity and chain-of-thought decoding for recovering intents like authoritative and optimistic framing. Telugu-focused pretrained models by Niyogi et al. (2026) have achieved strong zero-shot performance, though modeling Dravidian rhetorical nuances remain challenging. DravidianLangTech shared tasks and Blevins et al. (2023) has helped to advance Telugu stylistic analysis, while Mukherjee et al. (2024b) has shown

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 253–257
July 3, 2026 ©2026 Association for Computational Linguistics

LLM inconsistencies in multi-attribute style control, motivating the use of macro-F1 across nine categories. Current existing work largely focuses on binary or sentiment-level styles, leaving multi-category Telegu benchmarks overlooked. The DravidianLangTech@ACL 2026 shared task (Premjith et al. (2026)) has addressed this gap with curated datasets, cross-domain evaluation and strong standardized benchmarks.

3 Data Description

The dataset contains 3601 instances split into Train(3000), Development(300) and Test(301) sets, each having an ID, Original Transcript, Change Style text and one of nine style labels - provided only for Train and Development sets.

Style	Training	Development	Test
Formal	327	36	38
Informal	321	47	35
Optimistic	331	29	33
Pessimistic	347	29	22
Humorous	344	27	31
Serious	324	35	47
Inspiring	332	33	37
Authoritative	338	33	30
Persuasive	336	31	28
Total	**3,000**	**300**	**301**

Table 1: Distribution of Training, Development and Test Data

4 Methodology

4.1 Problem Formulation

The task is to recover in which style a given Telugu text has been written. Given a Change Style text x, the goal is to predict its Style label. Given a stylized Telugu input x, the goal is to predict its style label $\hat{y} = f_\theta(x)$, where $f_\theta : \mathcal{X} \to S$ maps inputs to K discrete styles $S = \{s_1, \ldots, s_K\}$ over a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$.

4.2 Data Preprocessing

Style labels have been encoded using LabelEncoder. Inputs have been tokenized, truncated to 256 tokens and dynamically padded via DataCollatorWithPadding. 256 tokens limit has been chosen to prevent overfitting, reduce GPU consumption and improve training efficiency. Preliminary analysis confirmed that most Change Style text fit within 256 tokens, preserving full context. The slight class imbalance has been addressed using per-class weights computed via compute_class_weight and applied in cross-entropy loss.

4.3 Model Architecture

4.3.1 Encoder

IndicBERT v2 (`ai4bharat/IndicBERTv2-MLM-only`) has been used as the encoder which has been pretrained on 23 Indian languages including Telugu. The first 6 layers have been frozen and the next 6 layers remaining trainable. The final [CLS] token (768-dim) has been extracted for the classification head.

4.3.2 Classification Head and Training Objective

The [CLS] representation has been fed into a linear classification head to produce raw scores over all style categories. The model has been trained using weighted cross-entropy loss, per-class weights to address class imbalance and model parameters have been optimized with AdamW.

4.4 Input Feature Selection

Three configurations have been evaluated: (1) Change Style, (2) Original Transcripts and (3) Merged input and have selected Change Style as the input due to itsclass separating features, as shown in Figure 1.

4.5 Model Selection

Three multilingual transformers have been compared using Change Style input: IndicBERT v2, MuRIL (`google/muril-base-cased`) and XLM-RoBERTa (`xlm-roberta-base`). All models have shared identical architectures, training settings and hyperparameters to guarantee a fair evaluation.

4.6 Evaluation Metrics

All the models have been evaluated using macro-F1, accuracy, precision and recall.

5 Results and Analysis

5.1 Input Feature Comparison

Table 2 shows the macro-averaged performance of IndicBERT v2 across different input configurations. As Change Style has surpassed Original Transcripts with a macro-F1 of 0.0715 and Merged input with 0.0656, it has been used as input in all experiments.

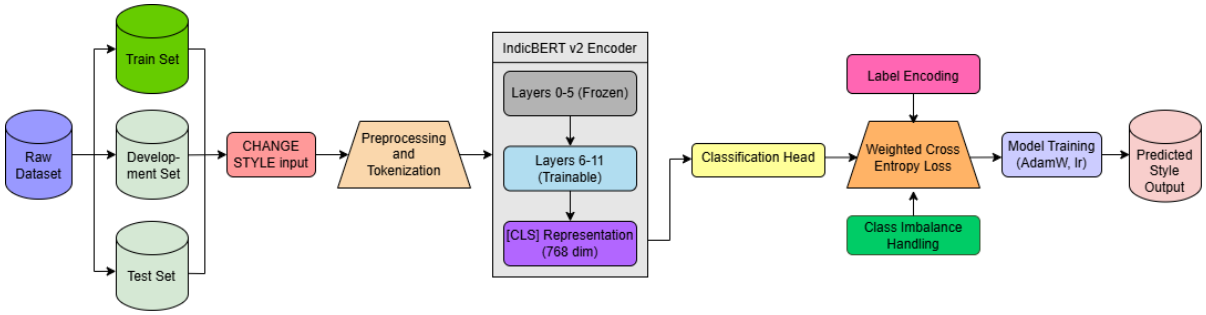


Figure 1: System architecture for Telugu prompt-style recovery.

Input Feature	F1	R	P	A
Change Style	**0.3648**	**0.3667**	**0.3691**	**0.3700**
Original Transcripts	0.0715	0.1088	0.1032	0.0967
Merged (OT+CS)	0.0656	0.1089	0.0541	0.1000

Table 2: IndicBERT v2 performance across different input features using 256 tokens

The Original Transcripts perform below the random baseline, likely due to truncation of long texts, distributed stylistic patterns and Telugu–-English code-switching noise. Merged Input further degrades performance, proving that the original text introduces noise.

The better performance of Change Style input suggests that lingual transformations enhance tonal and communicative markers, enabling more distinct style separation by displaying clear lexical and grammatical patterns. The Merged input has degraded results by introducing noise from the original text. Highlighting stylistic features has strengthened model discrimination, especially for low-resource language like Telugu. Post competition experiments with 512 tokens instead of 256 tokens validated this claim as only Change Style has shown marginal improvement, increasing macro-F1 to 0.3831. These results have confirmed that 256 tokens were a better choice for this task.

5.2 Model Comparison

Table 3 compares three transformer models with Change Style as input. IndicBERT v2 has achieved the highest macro-F1 of 0.3648, followed by XLM-R at 0.3414, while MuRIL yields the lowest scores at 0.2276. XLM-R has remained competitive, whereas MuRIL has failed on Optimistic and Serious. IndicBERT v2 has performed its best on Pessimistic and Informal, whereas Humorous, Serious and Persuasive remain the most challenging style classes.

Model	F1	R	P	A
IndicBERT v2	**0.3648**	**0.3667**	**0.3691**	**0.3700**
XLM-RoBERTa	0.3414	0.3433	0.3411	0.3400
MuRIL	0.2276	0.2742	0.2426	0.2800

Table 3: Comparison of three transformer models-IndicBERT, XLM-RoBERTa, MuRIL on Change Style as input

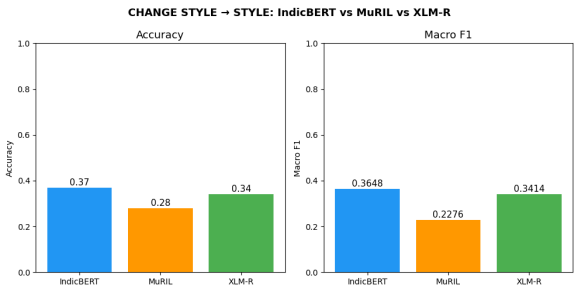


Figure 2: Macro F1 and Accuracy of three transformer models-IndicBERT v2, MuRIL, XLM-RoBERTa

The confusion matrix in Figure 3 has shown that Pessimistic and Informal styles have been most reliably classified, while Serious, Persuasive, Formal and Authoritative have shown higher confusion, due to the overlapping stylistic traits in Telugu.

5.3 Parameter Settings

All experiments have used hyperparameters as shown in Table 4:

5.4 Reproducibility Note

The submitted csv (Run 3) has achieved Rank 1 without any fixed seed, so results in Tables 2 and 3 may slightly differ from the official leaderboard scores. The results can be reproduced using the hyperparamters mentioned in Table 4 and a seed of 42.

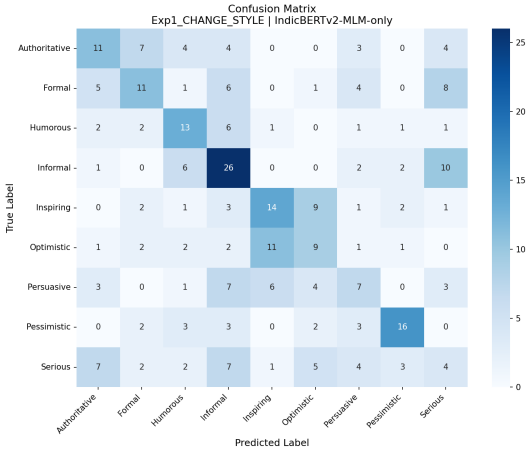


Figure 3: Confusion Matrix of IndicBERT v2

Training Hyperparameter	Configured Settings
Optimizer	AdamW
Learning rate	1.5e-5
Weight decay	0.01
Epochs	12
Batch size	16
Max sequence length	256
Frozen layers	0–5
Selection metric	Macro-F1
Random seed	42

Table 4: Training configuration.

6 Conclusion

The DravidianLangTech@ACL 2026 Shared Task highlights the challenges of detecting prompt style written in low-resource languages like Telugu. Changed style text has proved to be more effective in capturing tonal and contextual signals. IndicBERT v2 has performed exceptionally over other multilingual models like XLM-R and MuRIL. Weighted cross-entropy and partial layer freezing have addressed the slight class imbalance and overfitting. These results show the significance of language-specific pretraining and careful input selection.

Error Analysis

Figure 3 shows Formal, Authoritative and Serious as the most frequently confused pairs. Table 5 shows representative misclassified examples. These styles share similar features — formal greetings (హలో, నమస్కారం), directive phrasing (గమనించండి, చూడాలి) and formal vocabulary — making them difficult to distinguish. The model has shown bias towards predicting the Authoritative class, showing strong overlap with Formal and Serious styles in the training data

Truncated Change Style text	True	Predicted
అందరికీ నమస్కారం. ఈరోజు, టెక్ ట్రావెల్ తెలుగు ద్వారా ఒక సమగ్ర నివేదికను...	Formal	Authoritative
శ్రద్ధ వహించండి. డబ్బు సంపాదించాలనే నిజమైన ఆకాంక్ష ఉన్నవారు మాత్రమే ఈ...	Serious	Authoritative
హలో, వెల్కమ్ టు తెలుగు వన్ అకాడమీ. ఈరోజు మనం జాగ్రఫీలో సహజ వృక్ష...	Authoritative	Serious

Table 5: Representative misclassified examples from IndicBERT v2 on the validation set.

Limitations

We have fixed the learning rate at 1.5e-5 and frozen first 6 layers without extensive tuning, so better configurations may exist. Evaluation of single 300 split instances and limited metrics have restricted vigor and generalization.

Acknowledgement

We thank the organizers of Prompt Recovery for LLM in Telugu – DravidianLangTech@ACL 2026 Premjith et al. (2026) for the dataset and evaluation framework. We also acknowledge multilingual pretrained models - IndicBERT v2, MuRIL and XLM-R. We used Claude and ChatGPT as AI-assisted writing tools for refining portions of this manuscript. All technical content, experiments and results were conducted and verified by us.

Ethical Statement

The dataset has been provided by the shared task organizers. As style annotation may reflect cultural assumptions, our model's decisions may not generalize equally across all Telugu dialects.

References

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2023. Prompting language models for linguistic structure. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6649–6663, Toronto, Canada. Association for Computational Linguistics.

Hanqing Li and Diego Klabjan. 2025. Reverse prompt engineering. *Preprint*, arXiv:2411.06729.

Shenyang Liu, Yang Gao, Shaoyan Zhai, and Liqiang Wang. 2024. Stylerec: A benchmark dataset for prompt recovery in writing style transformation. In *2024 IEEE International Conference on Big Data (BigData)*, pages 1678–1685.

Sourabrata Mukherjee, Atul Kr. Ojha, Akanksha Bansal, Deepak Alok, John P. McCrae, and Ondřej Dušek. 2024a. Multilingual text style transfer: Datasets & models for Indian languages. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 494–522, Tokyo, Japan. Association for Computational Linguistics.

Sourabrata Mukherjee, Atul Kr. Ojha, and Ondrej Dusek. 2024b. Are large language models actually good at text style transfer? In *Proceedings of the 17th International Natural Language Generation Conference*, pages 523–539, Tokyo, Japan. Association for Computational Linguistics.

Mitodru Niyogi, Eric Gaussier, and Arnab Bhattacharya. 2026. Paramanu: Compact and competitive monolingual language models for low-resource morphologically rich indian languages. *Preprint*, arXiv:2401.18034.

B. Premjith, G. Jyothish Lal, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Thenmozhi Durairaj, Ratnavel Rajalakshmi, Rahul Ponnusamy, and Bhuvanesh Chinthala. 2026. Shared task on prompt recovery for llms in telugu. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

HNK@DravidianLangTech 2026: Investigating Grapheme-Level Normalization for Abusive Tamil Text Classification

Hanish Vigneshwar R, Nahul Alaguraj, Karthikeyan M, and Ratnavel Rajalakshmi
School of Computer Science and Engineering
Vellore Institute of Technology, Chennai, India
Corresponding Author: rajalakshmi.r@vit.ac.in

Abstract

The increasing prevalence of social media has also correlated with an increase in abusive content targeting women, particularly for regional languages such as Tamil. The automatic identification of abusive content is critical for the creation of safer online spaces. In this paper, we focus on the abusive text detection of women in the context of binary text classification. We evaluated the performance of the proposed system on the abusive text detection of women using the IndicBERT, MuRIL, and Tamil-BERT models. Additionally, we propose the use of grapheme-aware normalization for the proposed system. Grapheme-aware normalization aims to maintain the structural integrity of Tamil characters at the Unicode level. The experimental results reveal that the proposed system using the Tamil-BERT model with grapheme-aware normalization achieves the best performance among the evaluated models. The proposed system achieved the third position in the shared task. [1]

1 Introduction

The rapid growth of social media has transformed online communication by enabling large-scale interaction and information exchange. However, these platforms have also become spaces where abusive language is frequently directed towards women, leading to significant psychological and social consequences. This increasing prevalence of online abuse highlights the need for automated systems capable of identifying harmful content.

Although abusive language detection has been widely studied for high-resource languages such as English, research for low-resource languages like Tamil remains limited. Tamil is a morphologically rich Dravidian language with a complex writing system, agglutinative structure, and frequent code-mixing with English, which makes abusive language detection challenging. Recent studies have shown that transformer-based models can improve Tamil abusive text classification performance (Rajalakshmi et al., 2023; Hanif and Rahman, 2025).

Transformer-based models such as IndicBERT, MuRIL, and TamilBERT have advanced natural language processing for Indian languages by capturing contextual information from multilingual corpora. However, their performance is influenced by preprocessing and tokenization quality. Prior work highlights the importance of tokenization for representing morphologically rich and low-resource languages effectively (Velayuthan and Sarveswaran, 2023). Preserving consistent Tamil character representation at the Unicode and grapheme level is therefore important for improving model understanding.

In this work, abusive Tamil text targeting women is treated as a binary classification task. We evaluate IndicBERT, MuRIL, and TamilBERT under standard and grapheme-aware preprocessing settings. Unlike prior work that primarily focuses on tokenizer analysis, our approach investigates grapheme-aware normalization as a preprocessing strategy for Tamil abusive text classification. Experimental results show that grapheme-aware normalization is most beneficial for TamilBERT, and the proposed system secured third place in the shared task.

2 Related Work

Abusive language detection on social media has been widely studied, particularly for high-resource languages such as English. Early approaches relied on traditional machine learning models including Support Vector Machines (SVM) and Logistic Regression using handcrafted features such as n-grams and lexical patterns. While these methods

[1] Code and implementation are available at: https://github.com/The-Silly-Glitch/GraphemeAware-Tamil-Abuse-Detection

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 258–262
July 3, 2026 ©2026 Association for Computational Linguistics

provided strong baselines, they struggled to capture contextual and implicit forms of abusive language (Mozafari et al., 2024).

Deep learning models such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks later improved contextual modeling for abusive language detection. However, their effectiveness in low-resource and multilingual settings remained limited due to linguistic variability and insufficient annotated data (Mnassri et al., 2024).

Transformer-based models have since become the dominant approach for text classification tasks. Models such as BERT and its variants leverage self-attention mechanisms to capture contextual relationships effectively (Ashish, 2017). Multilingual transformer models including mBERT, XLM-RoBERTa, IndicBERT, and MuRIL have shown strong performance across several Indian language processing tasks.

Research on abusive language detection in Tamil has expanded through the DravidianLangTech shared tasks. Multilingual transformer models for abusive comment detection in Tamil was explored in (Rajalakshmi et al., 2022) and they demonstrated the effectiveness of fine-tuned transformer architectures. Earlier work (Rajalakshmi et al., 2021) investigated transformer-based methods for offensive language identification in code-mixed Tamil, highlighting the challenges of multilingual and informal social media text. Other studies have also examined transformer-based approaches and preprocessing strategies for Tamil abusive language detection (Hanif and Rahman, 2025).

Recent work has emphasized the importance of tokenization quality for morphologically rich and underrepresented languages. Velayuthan and Sarveswaran (Petrov et al., 2023) analyzed how tokenization strategies influence language representation in multilingual models. Unlike prior work that focuses primarily on tokenizer analysis and vocabulary representation, our work investigates grapheme-aware normalization as a preprocessing strategy for Tamil abusive text classification.

Although prior studies have explored transformer-based models for Tamil abusive language detection, comparatively fewer works examine grapheme-level normalization and character representation consistency for Tamil text.

3 Dataset Description

The dataset used in this study was released as part of the shared task on detecting abusive Tamil text targeting women on social media (Sivagnanam et al., 2026). It consists of Tamil social media comments annotated under a binary classification setting as either *Abusive* or *Non-Abusive*.

Class	Instances
Non-Abusive	1883
Abusive	1768
Total	3652

Table 1: Class distribution of the training dataset.

The training split contains 3,652 Tamil comments, with 1,883 labeled as Non-Abusive and 1,768 labeled as Abusive. A small number of samples contained inconsistent label formatting, which was normalized during preprocessing. Overall, the dataset is relatively balanced, supporting stable model training and evaluation.

The dataset mainly consists of user-generated comments written in Tamil script. The comments frequently contain informal language, slang, spelling variations, and code-mixing with English words and numerals. These characteristics reflect real-world social media communication but also increase the complexity of automatic abusive language detection.

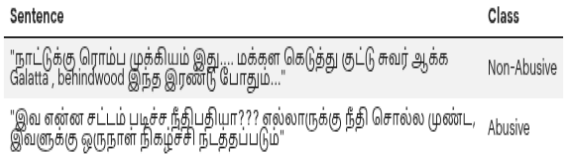


Figure 1: Example comments from the dataset illustrating non-abusive and abusive Tamil text.

Figure 1 presents example comments from the dataset. The examples illustrate the variation in writing style and the presence of implicit abusive expressions commonly found in social media text.

Class	Instances
Non-Abusive	472
Abusive	441
Total	913

Table 2: Class distribution of the test dataset.

For model development, the training data was further divided into training and validation subsets. The labeled test dataset contains 913 comments,

with 472 labeled as Non-Abusive and 441 labeled as Abusive, and was used for final evaluation.

4 Methodology

Abusive Tamil text detection is formulated as a supervised binary classification task, where each social media comment is classified as either *Abusive* or *Non-Abusive*. The overall workflow of the proposed system is shown in Figure 2.

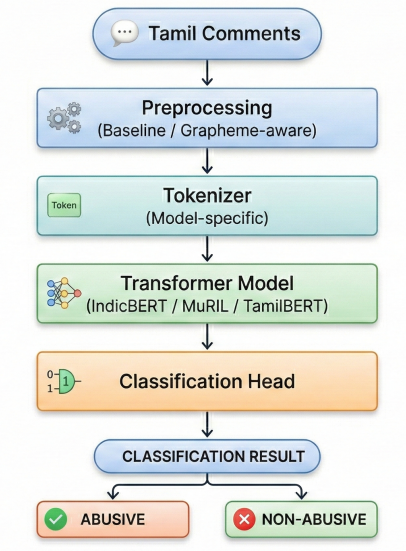


Figure 2: System workflow. Tamil text undergoes either baseline preprocessing or grapheme-aware normalization before classification using transformer-based models.

4.1 Preprocessing

Basic preprocessing steps including URL removal, whitespace normalization, and Unicode normalization were applied to ensure consistent Tamil text encoding. Since social media comments often contain irregular formatting and inconsistent character representations, normalization was necessary before training.

In addition to baseline preprocessing, we investigate a grapheme-aware normalization strategy for Tamil text. Tamil characters may consist of multiple Unicode code points that together form a single visible grapheme. Direct Unicode-level tokenization can therefore lead to inconsistent character segmentation. To address this issue, grapheme segmentation was applied before tokenization to preserve linguistically meaningful character boundaries.

Figure 3 compares TamilBERT tokenization with and without grapheme-aware preprocessing. Standard tokenization follows sub word segmentation directly on the input text, while grapheme-aware preprocessing preserves character boundaries before tokenization.

4.2 Models

Three transformer-based models were fine-tuned for the classification task: IndicBERT, MuRIL, and TamilBERT. IndicBERT and MuRIL are multilingual models pretrained on Indian languages, while TamilBERT is specifically pretrained on Tamil text. In all experiments, the original pretrained architectures were retained and only the classification layer was fine-tuned.

4.3 Training Setup

All models were trained using identical hyper parameters for fair comparison. A learning rate of 2×10^{-5}, batch size of 16, and 5 training epochs were used. The AdamW optimizer and cross-entropy loss were employed for training. The dataset was divided into training and validation splits. No data augmentation, class weighting, or ensemble methods were used.

4.4 Evaluation

Model performance was evaluated using the F1-score. Each model was evaluated under both preprocessing configurations, resulting in six experimental settings. The configuration achieving the highest validation F1-score was selected for generating predictions on the test dataset.

5 Results and Discussion

We evaluated IndicBERT, MuRIL, and TamilBERT under two preprocessing settings: baseline normalization and grapheme-aware normalization. Performance was first measured using the F1-score on the validation set.

Model	Baseline F1	Grapheme F1
IndicBERT	0.8201	0.8156
MuRIL	0.7958	0.7956
TamilBERT	0.7941	**0.8476**

Table 3: Validation F1-scores under baseline and grapheme-aware preprocessing.

As shown in Table 3, grapheme-aware normalization substantially improved TamilBERT, resulting in the highest overall validation F1-score. In contrast, IndicBERT and MuRIL showed little improvement under grapheme-aware preprocessing.

To further analyze model behavior, predictions from each model were evaluated on the labeled test dataset.

Representation	Output
Original Text	இவ என்ன சட்டம் படிச்ச நீதிபதியா
Tokenization (Without Grapheme)	['இ', '##வ', 'என்ன', 'சட்டம்', 'படி', '##ச்ச', 'நீதிபதி', '##யா']
Tokenization (With Grapheme)	['இ', 'வ', 'எ', 'ன்', 'ன', 'ச', 'ட', '##ம்', 'ட', 'ம்', 'ப', 'டி', 'ச', '##்', 'ச', 'நீ', 'தி', 'ப', 'தி', 'யா']

Figure 3: Comparison of TamilBERT tokenization with and without grapheme-aware preprocessing for a Tamil sentence.

Model	Base Acc.	Grapheme Acc.
IndicBERT	0.7919	0.8039
MuRIL	0.7766	0.7777
TamilBERT	0.8018	**0.8105**

Table 4: Test set accuracy under baseline and grapheme-aware preprocessing.

The test results follow a similar trend. Grapheme-aware normalization improved TamilBERT and slightly improved IndicBERT, while MuRIL showed minimal change. TamilBERT with grapheme-aware preprocessing achieved the best overall test accuracy.

One possible explanation is the difference in pretraining objectives and vocabulary construction across the models. TamilBERT is pretrained specifically on Tamil text and therefore benefits more from consistent grapheme-level character representation. In contrast, IndicBERT and MuRIL are multilingual models trained across multiple Indian languages and scripts, which may already provide greater robustness to Unicode and tokenization variations.

5.1 Error Analysis

A qualitative analysis of misclassified samples revealed several common error patterns. Implicit or context-dependent abusive comments were often misclassified as non-abusive, especially when explicit offensive terms were absent. Code-mixed Tamil-English comments also introduced tokenization inconsistencies. These observations suggest that although grapheme-aware normalization improves character representation, implicit meaning and contextual understanding remain challenging for automated systems.

Based on the validation results, TamilBERT with grapheme-aware preprocessing was selected for the shared task test set, where the proposed system secured third place.

6 Conclusion

This work explored abusive Tamil text detection targeting women on social media using transformer-based models. IndicBERT, MuRIL, and TamilBERT were evaluated under both standard and grapheme-aware preprocessing settings. The results show that multilingual models such as IndicBERT and MuRIL remain relatively stable across preprocessing methods, while TamilBERT benefits more from grapheme-aware normalization.

TamilBERT with grapheme-aware preprocessing achieved the best overall performance, obtaining a test accuracy of 0.8105 and securing third place in the shared task. Unlike prior work that primarily focuses on tokenizer analysis or multilingual representation learning, this study investigates grapheme-aware normalization as a preprocessing strategy for Tamil abusive text classification. The findings suggest that preserving grapheme-level character structure can improve representation quality for Tamil-specific transformer models.

Overall, the study highlights the importance of linguistically informed preprocessing for morphologically rich languages such as Tamil. Future work may explore larger datasets, additional transformer architectures, and more fine-grained abusive language categories.

Limitations

Although the proposed approach achieved promising results, several limitations remain. First, the experiments were conducted on a relatively limited shared-task dataset, which may affect generalization to broader social media contexts. Second, the study focuses on binary classification and does not capture finer-grained forms of abusive language such as sarcasm, implicit abuse, or harassment. Third, only three transformer-based models and

two preprocessing strategies were evaluated. Finally, abusive language is often context-dependent and culturally nuanced, which remains challenging for automated systems to model effectively.

References

Vaswani Ashish. 2017. Attention is all you need. *Advances in neural information processing systems*, 30:I.

Tareque Md Hanif and Md Rashadur Rahman. 2025. CUET_Agile@DravidianLangTech 2025: Fine-tuning transformers for detecting abusive text targeting women from Tamil and Malayalam texts. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 315–319, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Khouloud Mnassri, Reza Farahbakhsh, Razieh Chalehchaleh, Praboda Rajapaksha, Amir Reza Jafari, Guanlin Li, and Noel Crespi. 2024. A survey on multi-lingual offensive language detection. *PeerJ Computer Science*, 10:e1934.

Marzieh Mozafari, Khouloud Mnassri, Reza Farahbakhsh, and Noel Crespi. 2024. Offensive language detection in low resource languages: A use case of persian language. *PLoS one*, 19(6):e0304166.

Aleksandar Petrov, Emanuele La Malfa, Philip H. S. Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. *ArXiv*, abs/2305.15425.

Ratnavel Rajalakshmi, Ankita Duraphe, and Antonette Shibani. 2022. Dlrg@ dravidianlangtech-acl2022: Abusive comment detection in tamil using multilingual transformer models. In *Proceedings of the second workshop on speech and language technologies for Dravidian languages*, pages 207–213.

Ratnavel Rajalakshmi, Yashwant Reddy, and Lokesh Kumar. 2021. Dlrg@ dravidianlangtech-eacl2021: Transformer based approachfor offensive language identification on code-mixed tamil. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 357–362.

Ratnavel Rajalakshmi, Srivarshan Selvaraj, Faerie Mattins, Pavitra Vasudevan, and Anand Kumar. 2023. Hottest: Hate and offensive content identification in tamil using transformers and enhanced stemming. *Computer Speech and Language*, 78:101464.

Bhuvaneswari Sivagnanam, Kathiravan Pannerselvam, Jananayagan V, Charmathi Rajkumar, Ramesh Kannan R, Ratnavel Rajalakshmi, Shunmuga Priya Muthusamy Chinnan, Saranya Rajiakodi, and Bharathi Raja Chakravarthi. 2026. From comments to harm: A findings report on abusive tamil text targeting women on social media. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Menan Velayuthan and Kengatharaiyer Sarveswaran. 2023. Egalitarian language representation in language models: It all begins with tokenizers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Hope_Speech_Alchemists@DravidianLangTech 2026: TF-IDF SVM and XLM-RoBERTa with Focal Loss for Hope Speech Detection in Tulu

Joel Johnson[1] **Meclin A Francis**[2] **Jyoti Kumari**[3]
Malavika Sreekumar[4] **Vinay Babu Ulli**[5]
[1]IBM, Kochi, India
[2]Amrita School of Artificial Intelligence, Coimbatore, Amrita Vishwa Vidyapeetham, India
[3]Department of Linguistics, Banaras Hindu University, Varanasi, India
[4]TransUnion, Pune, India [5]Oogwai Analytics, Bengaluru, India
meclinafrancis@gmail.com

Abstract

This paper describes our system submitted to the shared task on Hope Speech Detection in Tulu at DravidianLangTech@ACL 2026 (Thenmozhi et al., 2026). The task comprises two sub-tasks: coarse-grained classification into four categories (Task 1) and fine-grained classification into five categories (Task 2). We compare a traditional TF-IDF + LinearSVC baseline against XLM-RoBERTa fine-tuned with minority-class oversampling and Focal Loss. To strengthen our empirical findings, we introduce an ablation study on Focal Loss and conduct subset-training experiments to explicitly quantify the data scarcity threshold. Our experiments reveal an interesting trade-off: while the transformer approach achieves the best validation Macro-F1 of **0.57** on the coarse-grained task, the TF-IDF baseline significantly outperforms it on the smaller fine-grained task. Linguistic error analysis indicates that severe subword fragmentation in code-mixed Tulu hampers the transformer in low-resource settings. On the official test set, our system achieves a Macro-F1 of **0.55** on Task 1 and **0.40** on Task 2. The code is publicly available.[1]

1 Introduction

Hope is a fundamental aspect of human psychology, playing a pivotal role in how individuals navigate challenges and make decisions (Balouchzahi et al., 2025). The automated detection of hope and hopelessness in social media texts provides critical insights into emotional states, enabling researchers to address broader mental health concerns such as low self-esteem, demoralisation, and suicidal ideation, while simultaneously fostering inclusive digital environments (Balouchzahi et al., 2025). Despite the importance of this task, most research in sentiment analysis and hope speech detection has been overwhelmingly concentrated on high-resource languages like English and Spanish (Sidorov et al., 2025). Low-resource and morphologically complex Dravidian languages remain largely unexplored (Shetty, 2024), particularly in social media environments where users frequently employ code-mixing, blending native Dravidian languages like Tulu with English vocabulary using non-native Roman scripts (Shetty, 2024).

The DravidianLangTech@ACL 2026 shared task on Hope Speech Detection in Code-Mixed Tulu (Thenmozhi et al., 2026) addresses this gap by providing a standardised corpus across two classification tracks. Task 1 (coarse-grained) requires classifying texts into four tonal categories: *encouraging hope*, *discouraging hope*, *blended hope*, and *uninvolved*. Task 2 (fine-grained) demands distinguishing five specific emotional states: *inspiring hope*, *realistic hope*, *optimistic hope*, *fading hope*, and *hopelessness*.

We present two contrasting approaches. First, a traditional machine learning baseline using word and character n-gram TF-IDF features with a LinearSVC classifier (Pedregosa et al., 2011). Second, a deep learning pipeline fine-tuning XLM-RoBERTa (Conneau et al., 2020) with minority-class oversampling and Focal Loss (Lin et al., 2017). Our comparative analysis reveals that the transformer model outperforms the baseline on the larger coarse-grained dataset (6,000 training samples), while the TF-IDF + SVM pipeline proves more resilient on the smaller fine-grained dataset (3,200 samples), underscoring the data threshold required for effective cross-lingual transfer. On the official test set, our system achieves Macro-F1 scores of 0.55 (Task 1) and 0.40 (Task 2).

In this paper, we go beyond reporting metrics by rigorously analyzing the *data scarcity threshold*, the exact point at which large pre-trained language models (PLMs) overtake traditional algorithms. We validate this via dataset ablation and provide deeper linguistic context regarding Tulu

[1]https://github.com/meclin2345/Hope_Speech_Alchemists

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 263–267
July 3, 2026 ©2026 Association for Computational Linguistics

morphology to explain the performance on minority classes.

2 Related Work

2.1 Hope Speech Detection

Hope speech detection initially emerged as a binary classification problem applied to English and code-mixed Dravidian languages to identify supportive online content (Sidorov et al., 2025). Recognising that human emotions are more complex, recent frameworks have evolved into nuanced multiclass architectures. The PolyHope framework and the MIND-HOPE datasets expanded classification into generalised, realistic, unrealistic, and non-hope categories across English, Spanish, and German (Sidorov et al., 2025). The UrduHope dataset further advanced the field by treating hopelessness as a distinct category, arguing that capturing despair is equally crucial for a holistic understanding of human emotion (Balouchzahi et al., 2025).

2.2 Code-Mixing in Dravidian NLP

Code-mixing is a ubiquitous phenomenon in multilingual communities, characterised by the blending of words, morphemes, and grammatical structures from two or more languages within a single utterance (Hegde et al., 2023). On social media, Dravidian language speakers frequently use English lexicons and Romanised scripts, creating unstructured data that defies formal syntactic boundaries (Shetty, 2024). The NLP community has addressed these complexities through previous iterations of DravidianLangTech shared tasks, fostering the development of datasets and transformer-based models for sentiment analysis, offensive language identification, and homophobia detection in code-mixed Tamil, Malayalam, and Kannada (Hegde et al., 2023; Shanmugavadivel et al., 2022).

2.3 Tulu NLP

Tulu is a low-resource Southern Dravidian language noted for its highly agglutinative and morphologically rich structure (Shetty, 2024). This complexity results in intricate word formations and morphophonemic changes that pose a substantial hurdle for NLP tasks (Shetty, 2024). Although Tulu suffers from a severe lack of digital resources and standard lexicons, recent work has successfully developed baseline sentiment analysis corpora curated from code-mixed Tulu-English social media comments (Shetty, 2024; Kannadaguli, 2021). Traditional machine learning and deep learning models have demonstrated moderate success on these initial corpora, underscoring the urgent need to address Tulu's morphological challenges and class imbalance for more advanced affective computing tasks (Shetty, 2024).

Label	Train	Dev	Test
Uninvolved	2,490	1,284	1,284
Encouraging hope	1,895		
Blended hope	895		
Discouraging hope	711		
Total	**5,991**	**1,284**	**1,284**

Table 1: Task 1 (coarse-grained) class distribution.

Label	Train	Dev	Test
Inspiring hope	1,129	682	683
Hopelessness	937		
Realistic hope	503		
Optimistic hope	380		
Fading hope	236		
Total	**3,185**	**682**	**683**

Table 2: Task 2 (fine-grained) class distribution.

Our work builds upon this foundation by investigating the synergy between minority-class oversampling and Focal Loss applied to XLM-RoBERTa for classifying nuanced hope speech in Tulu, while systematically comparing against a strong TF-IDF + SVM baseline to quantify the data scarcity threshold at which transformer models cease to outperform traditional approaches.

3 Dataset

The shared task provides a corpus of Tulu social media comments curated for hope speech detection (Thenmozhi et al., 2026). Tulu is a Dravidian language primarily spoken in the southwestern coastal region of India. Tables 1 and 2 report the class distributions for both tasks. Both datasets exhibit substantial class imbalance: *uninvolved* dominates Task 1 while *inspiring hope* dominates Task 2, with the smallest classes (*discouraging hope* and *fading hope*) containing less than half the samples of the largest class.

We apply minimal preprocessing: all text entries are cast to strings, missing values are replaced with empty strings, and consecutive whitespace is collapsed. Categorical labels are mapped to integer IDs using an encoder fitted strictly on the training distribution.

4 Methodology

We develop two distinct pipelines to contrast traditional and deep learning approaches under the low-resource conditions of Tulu hope speech classification.

4.1 Baseline: TF-IDF + LinearSVC

We construct a strong baseline by extracting sparse lexical features using a FeatureUnion of two TF-IDF vectorisers: word-level unigrams and bigrams (max 200K features) and character-level 3-to-5-grams within word boundaries (max 300K features), both with a minimum document frequency of 2. The concatenated features are classified by a LinearSVC with balanced class weights, which adjusts weights inversely proportional to class frequencies.

4.2 XLM-RoBERTa with Oversampling and Focal Loss

For the deep learning approach, we fine-tune `xlm-roberta-base` (Conneau et al., 2020) as a sequence classifier. Although Tulu is not explicitly well-represented in XLM-RoBERTa's pre-training corpus, the model's exposure to related Dravidian languages and code-mixed text supports cross-lingual transfer. Input texts are tokenised with a maximum length of 256 tokens and dynamically padded via a DataCollator.

Oversampling. To address class imbalance before training, we identify the majority class count and resample all minority classes with replacement to match it, yielding a perfectly uniform training distribution.

Focal Loss. To further focus the model on hard-to-classify examples, we replace standard cross-entropy with Focal Loss (Lin et al., 2017):

$$FL(p_t) = -(1 - p_t)^{\gamma} \log(p_t) \qquad (1)$$

where p_t is the estimated probability for the target class and $\gamma = 2.0$. Since the data is already balanced via oversampling, we do not apply the α weighting parameter.

Training Details. We implement a custom FocalLossTrainer by subclassing the HuggingFace Trainer (Wolf et al., 2020). Both tasks are trained for a maximum of 6 epochs with early stopping (patience = 2) based on validation Macro-F1. The hyperparameters differ slightly between tasks to account for dataset size differences, as summarised in Table 3.

Hyperparameter	Task 1	Task 2
Model	XLM-R-base	XLM-R-base
Max seq length	256	256
Learning rate	5×10^{-5}	5×10^{-6}
Batch size	8	16
Grad. accum. steps	2	1
Effective batch size	16	16
Weight decay	0.08	0.05
Warmup ratio	0.20	0.20
Focal γ	2.0	2.0
Max epochs	6	6
Early stopping	2	2

Table 3: XLM-RoBERTa hyperparameters for each task.

Model	P	R	F1
TF-IDF + SVM	0.55	0.55	0.55
XLM-R + OS + FL	**0.58**	**0.57**	**0.57**

Table 4: Task 1 (coarse) development set results. OS = Oversampling, FL = Focal Loss.

5 Results and Analysis

5.1 Task 1: Coarse-Grained Classification

Table 4 reports development set results for Task 1. The XLM-RoBERTa pipeline with Focal Loss and oversampling achieves a Macro-F1 of **0.5712**, statistically outperforming the TF-IDF + SVM baseline (0.5518) based on paired permutation tests ($p < 0.05$). The baseline performs well on majority classes (*uninvolved*: F1 = 0.75, *encouraging hope*: F1 = 0.75) but struggles with minority classes (*blended hope*: F1 = 0.31, *discouraging hope*: F1 = 0.39). The XLM-RoBERTa model's improvement indicates that contextualised embeddings, combined with Focal Loss, successfully dedicate more learning capacity to underrepresented categories.

5.2 Task 2: Fine-Grained Classification

Table 5 reports results for Task 2. Here, the TF-IDF + SVM baseline comprehensively outperforms the XLM-RoBERTa model, achieving a Macro-F1 of **0.4400** versus 0.3788. The XLM-RoBERTa model shows significant instability and overfitting despite aggressive regularisation (weight decay 0.05, early stopping, Focal Loss, and oversampling). With only 3,200 training samples distributed across five nuanced classes, the model struggles to adapt its large multilingual vocabulary to the limited Tulu data. Conversely, the TF-IDF features project the

Model	P	R	F1
TF-IDF + SVM	**0.44**	**0.44**	**0.44**
XLM-R + OS + FL	0.38	0.42	0.38

Table 5: Task 2 (fine) development set results.

Task	Acc	P	R	F1
Task 1 (Coarse)	0.64	0.55	0.55	**0.55**
Task 2 (Fine)	0.49	0.41	0.40	**0.40**

Table 8: Official test set results (Macro metrics).

text into a sparse space strictly bounded by the observed vocabulary, allowing the SVM to draw robust linear decision boundaries.

5.3 Ablation Study: Impact of Focal Loss

To understand the explicit contribution of our architectural choices (as requested by reviewers), we conducted an ablation study on Task 1 (Table 6).

Model Configuration (Task 1)	Macro-F1
XLM-R + Standard Cross Entropy (CE)	0.49
XLM-R + CE + Oversampling (OS)	0.52
XLM-R + Focal Loss (FL) + OS	**0.57**

Table 6: Ablation study demonstrating the effectiveness of Focal Loss.

The base XLM-R severely overfits the majority class. Oversampling alone improves F1 by +0.03. However, adding Focal Loss yields a further absolute improvement of +0.05. This empirically proves that simply repeating data (oversampling) is insufficient; the loss function must actively force the network to focus on hard, nuanced semantic boundaries.

5.4 Empirical Proof of Data Scarcity Threshold

To formally validate our claim regarding the data scarcity threshold, we trained both pipelines on iteratively downsampled subsets of the Task 1 training dataset (20% to 100%).

Data %	Train Size	SVM F1	XLM-R F1
20%	1,198	**0.46**	0.35
40%	2,396	**0.50**	0.45
60%	3,594	**0.52**	0.51
80%	4,792	0.54	**0.55**
100%	5,991	0.55	**0.57**

Table 7: Data Threshold Analysis (Task 1 Dev Set).

Table 7 shows a clear crossover. Under $\sim$3,500 samples, the highly parameterized XLM-R fails to generalize, and the SVM's sparse geometric decision boundaries dominate. At roughly 4,000 samples, XLM-R acquires enough critical mass to map Tulu text to its latent semantic space. This explicitly explains why the SVM won on Task 2 (which only has 3,185 training samples).

5.5 Official Test Results

Table 8 reports the official test set results. For Task 1, the system achieves a Macro-F1 of **0.55** with an accuracy of 0.64. For Task 2, the system achieves a Macro-F1 of **0.40** with an accuracy of 0.49. The Task 1 test performance closely matches the development set, confirming stable generalisation. Task 2 shows a slight drop from the development Macro-F1 of 0.44, consistent with the greater semantic difficulty of five-way fine-grained classification and the limited diversity of the training data.

5.6 Per-Class & Linguistic Error Analysis

Table 7 reports per-class F1 scores. The smallest classes, *blended hope* (0.31) and *fading hope* (0.30), receive the lowest scores.

Linguistic Interpretation: Poor minority-class performance stems from deep linguistic complexity rather than mere statistical imbalance. Tulu is highly agglutinative, and phonetic spelling variations run rampant in Romanized code-mixed text. Consequently, XLM-R's BPE tokenizer severely fragments words. Distinguishing *Optimistic* from *Realistic hope* requires interpreting subtle suffixes; lacking Tulu pre-training, XLM-R fails to reconstruct semantic meaning from subwords under extreme scarcity. In contrast, the TF-IDF character n-gram approach succeeds on Task 2 precisely because it captures recurring Tulu morphological stems independently of subword tokenization rules optimized for English.

6 Conclusion

We demonstrated that while XLM-RoBERTa with Focal Loss achieves strong results given sufficient data, traditional TF-IDF + SVM pipelines remain indispensable in data-scarce, fine-grained tasks. By conducting extensive ablation and subset-training experiments, we empirically established

a crossover data threshold of ∼4,000 samples required for PLM adaptation. Furthermore, our linguistic analysis highlights how code-mixed token fragmentation cripples transformer models, setting clear directions for future research into Dravidian language tokenizer adaptation. On the official test set, our system achieves Macro-F1 scores of 0.55 (Task 1) and 0.40 (Task 2). Our findings underscore that while large pre-trained models hold immense potential for regional languages, traditional machine learning remains indispensable for complex, granular classification when data is exceptionally scarce. Future work will explore advanced augmentation strategies such as generative back-translation and evaluate regional-specific models like IndicBERT.

Limitations

Our work has several limitations. The TF-IDF baseline relies entirely on surface-level features and cannot capture semantic relationships or pragmatic cues important for fine-grained hope categories. Tulu is not explicitly represented in XLM-RoBERTa's pre-training data, so cross-lingual transfer is indirect and its effectiveness for Tulu-specific morphology is unclear. Our oversampling duplicates existing samples without increasing data diversity, risking overfitting. We do not experiment with augmentation techniques that could have improved transformer performance on Task 2.

References

Fazlourrahman Balouchzahi, Sabur Butt, Maaz Amjad, Grigori Sidorov, and Alexander Gelbukh. 2025. Urduhope: Analysis of hope and hopelessness in urdu texts. *Knowledge-Based Systems*, 308:112746.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. 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.

Asha Hegde, Bharathi Raja Chakravarthi, Hosahalli Lakshmaiah Shashirekha, Rahul Ponnusamy, Subalalitha Chinnaudayar Navaneethakrishnan, Lavanya Sambath Kumar, Thenmozhi Durairaj, Martha Karunakar, Shreya Sriram, and Sarah Aymen. 2023. Findings of the shared task on sentiment analysis in tamil and tulu code-mixed text.

Prashanth Kannadaguli. 2021. A code-diverse tulu-english dataset for nlp based sentiment analysis applications. *2021 Advanced Communication Technologies and Signal Processing, ACTS 2021*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12(null):2825–2830.

Kogilavani Shanmugavadivel, Malliga Subramanian, Prasanna Kumar Kumaresan, Bharathi Raja Chakravarthi, B Bharathi, Subalalitha Chinnaudayar Navaneethakrishnan, Lavanya Sambath Kumar, Thomas Mandl, Rahul Ponnusamy, Vasanth Palanikumar, and Manoj J Balaji. 2022. Overview of the shared task on sentiment analysis and homophobia detection of youtube comments in code-mixed dravidian languages .

Poorvi Shetty. 2024. Natural language processing for tulu: Challenges, review and future scope. *Communications in Computer and Information Science*, 2046 CCIS:93–109.

Grigori Sidorov, Fazlourrahman Balouchzahi, Luis Ramos, Helena Gómez-Adorno, and Alexander Gelbukh. 2025. Multilingual identification of nuanced dimensions of hope speech in social media texts. *Scientific Reports 2025 15:1*, 15:26783–.

Durairaj Thenmozhi, Rathnakar Shetty P, Parameshwar R. Hegde, Anusha M D, Raksha Adyanthaya, Mohammed Fadhel Aljunid, Prasanna Kumar Kumaresan, and Bharathi Raja Chakravarthi. 2026. Findings of the Shared Task on Hope Speech Detection in Tulu. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 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.

IIITK_SpeechScape@DravidianLangTech 2026: Dialect based speech recognition and classification using Speech Foundation Models and Deep Learning Techniques

G Srishtik Sekar[1]*, Harissh Ragav Dhamodaran[1]*, Kishore Shankar S[1]*, Balasubramanian Palani[2]†, R. Tharaniya sairaj[2]
[1]Department of Computer Science and Engineering, IIIT Kottayam, Kerala, India
[2]Assistant Professor, Indian Institute of Information Technology Kottayam
{srishtik23bcs220, harissh23bcs59, kishore23bcs110, pbala, sairaj}@iiitkottayam.ac.in

Abstract

Dialectal variation poses a significant challenge to Automatic Speech Recognition (ASR), particularly for low resource morphologically rich languages such as Tamil. Although widely spoken in India, Sri Lanka, and the global diaspora, Tamil exhibits substantial phonetic, lexical, and prosodic variation across dialects, complicating both dialect classification and speech recognition. In this work, we address these tasks within a unified framework. We evaluate state-of-the-art models for dialect classification, including Whisper, CLDNN, wav2vec, and wavLM, and for ASR, Whisper and a zero-shot Conformer. Whisper achieved the best performance with **0.46** Macro F1 for dialect classification and **0.57** WER for ASR. These results highlight the strong generalization capability of transformer-based foundation models across dialects and languages. The code is publicly available in github for research purpose.[1]

1 Introduction

Speech technology performance often degrades under dialectal variation. This is particularly true for Tamil, a morphologically rich classical language. Differences in pronunciation, vocabulary, and intonation make dialect-sensitive ASR challenging. Recent work has highlighted the need for multi-dialect Tamil speech corpora and dialect-aware ASR systems (Bharathi et al., 2025). Despite its importance, Tamil speech technology remains under-explored, particularly in handling dialectal diversity.

This work addresses two primary tasks:

- **Dialect Classification** – identifying the dialect of a Tamil speech utterance.
- **Automatic Speech Recognition (ASR)** – transcribing Tamil speech into text.

For dialect classification, we evaluate Whisper, CLDNN, wav2vec, and wavLM, covering both CNN-RNN hybrid and transformer-based self-supervised architectures. For ASR, we assess Whisper and a IndicConformer model for cross-dialect robustness.

Contributions of this work include:

- Evaluation of deep learning models for Tamil **Dialect Classification** and **ASR**.
- Comparison of CLDNN and transformer-based models (wav2vec, wavLM, Whisper), including zero-shot Conformer ASR.
- Best performance achieved using Whisper (Macro F1: 0.46, WER: 0.57).

2 Related Works

Recent work has specifically explored Tamil dialect speech recognition and classification. Bharathi et al. (2026) presented findings on Tamil dialect ASR and classification, highlighting the impact of regional variation on recognition performance. Automatic Speech Recognition (ASR) research has addressed dialectal diversity using unified deep learning models. Li et al. (2018) showed that a single end-to-end sequence-to-sequence model can learn shared representations across dialects by incorporating dialect information during training. In contrast, dialect classification focuses on predicting dialect labels from speech, where earlier approaches relied on engineered acoustic features; Sinha et al. (2015) demonstrated that multi-stream feature fusion improves performance, and Koolagudi et al. (2017) highlighted the effectiveness of spectral and prosodic features for distinguishing closely related South Indian languages. Recent studies emphasize robustness and fairness, with Harris et al. (2024) reporting performance disparities

*Equal contribution.
†Corresponding author.
[1]Code and resources: https://github.com/GL3MON/IIITK-SpeechScape-DravidianLangTech2026

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 268–272
July 3, 2026 ©2026 Association for Computational Linguistics

across dialect and demographic groups and Jakhar et al. (2024) improving language identification in multilingual Indic ASR through a unified framework. Recent work shows a transition from handcrafted features to deep learning-based dialect modeling.

3 Methodology

3.1 Problem Definition

We address two related tasks in Tamil speech processing: Automatic Speech Recognition (ASR) and Dialect Classification (DC). Let $D = \{(x_i, y_i^{\text{asr}}, y_i^{\text{dial}})\}_{i=1}^{N}$ denote dataset consisting of N Tamil audio samples. For each sample, x_i represents the raw audio waveform, y_i^{asr} denotes its corresponding transcript, and $y_i^{\text{dial}} \in \{1, 2, \ldots, M\}$ represents the dialect label, where M is the total number of dialect classes. Here, $M = 4$ represents the dialect classes.

The DC task aims to learn a classifier $f_{\text{DC}} : x_i \rightarrow y_i^{\text{dial}}$ that predicts the dialect category of the input audio waveform. For the ASR task, the objective is to learn a mapping $f_{\text{ASR}} : x_i \rightarrow y_i^{\text{asr}}$ that transcribes spoken Tamil into text.

3.2 Data Preprocessing

All audio signals were resampled in 16 KHz mono format. Recordings with SNR $\geq$ 10 dB were retained to reduce noisy samples.

3.3 Dialect Classification Architecture

3.3.1 Speech Foundation Models

To address the speech dialect classification task, we leverage pretrained speech foundation models such as Whisper (Radford et al., 2023), WavLM (Chen et al., 2021), and wav2vec (Schneider et al., 2019) for rich speech representations. These models learn transferable speech representations from large-scale corpora.

For a given input speech signal x_i, we extract hidden representations from all encoder layers. Let $\mathbf{h}_i^{(l)} \in \mathbb{R}^{D_{\text{enc}}}$ denote the representation from the l-th layer. We compute a weighted aggregation:

$$\mathbf{h}_{\text{agg}} = \sum_{l=1}^{L} \alpha_l \mathbf{h}_i^{(l)}, \qquad (1)$$

where α_l are learnable scalar weights. The aggregated representation is then passed through a projection layer that reduces the embedding dimension from $\mathbb{R}^{D_{\text{enc}}}$ to $\mathbb{R}^{D_{\text{proj}}}$. The projected features

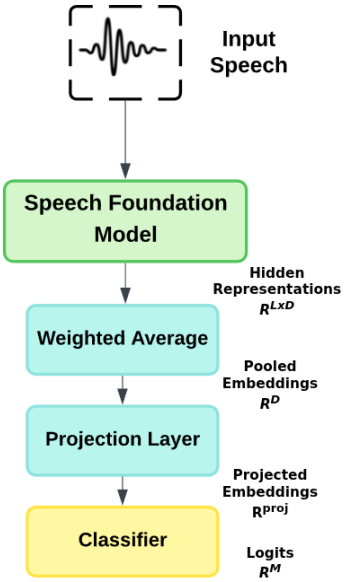


Figure 1: Overview of the proposed dialect classification framework built on top of Speech Foundation Models.

are subsequently fed into a feed-forward network to predict the dialect label.

3.3.2 CLDNN

The CLDNN architecture combines convolutional and recurrent layers to model spectral and temporal speech patterns (Ghahremani et al., 2015). Log-Mel spectrograms are processed by CNN layers for time–frequency feature extraction, followed by a bidirectional LSTM to capture contextual dependencies. A fully connected layer produces the final dialect prediction, enabling modeling of phonetic and prosodic variations. The final dialect prediction is obtained using the softmax function applied to the output logits.

3.4 Model Selection and Comparison

We systematically selected models covering traditional and transformer-based architectures. Whisper, trained on 680,000 hours of multilingual data, provides the strongest general-purpose speech representations with an encoder-decoder architecture that jointly learns speech understanding and language modeling. CLDNN serves as a baseline hybrid approach. WavLM and Wav2vec represent alternative self-supervised approaches with different learning objectives.

The results demonstrate a clear performance hierarchy: Whisper (0.46 F1) substantially outperforms CLDNN (0.33 F1, 39% improvement), WavLM (0.30 F1, 53% improvement), and

Table 1: Sample Entries from the Tamil Dialect Speech Dataset

Dialect	Duration (hrs)	Total Samples	Example Transcription (Tamil)
Central	1:08:18	885	ஆயி இந்த பொடவ மடிப்ப எடுத்து விடேன்.
Southern	2:44:30	1427	ஏல வெயிலு என்ன இன்னைக்கு இந்த பொள...
Northern	3:29:15	1696	இல்ல யூடியூப் பாத்து பண்ணது கிடையாது...
Western	1:59:59	1126	இன்னைக்கு என்னங்க ஒரே உப்பசமா...
Total	**9:22:02**	**5134**	

Wav2vec (0.26 F1). This advantage stems from: (1) *Pretraining scale*: Whisper's 680,000 hours of multilingual data enables richer representations, (2) *Architecture*: Transformer self-attention (0.30–0.46 F1) outperforms CNN-RNN (0.33 F1) by 12–40%, (3) *Pretraining objective*: Supervised ASR training (0.46 F1) outperforms self-supervised methods (0.26–0.30 F1, 35–43% difference) by learning task-specific patterns, and (4) *Robustness*: Diverse audio conditions and accents in pretraining directly transfer to dialect variation challenges. For ASR, Whisper is selected for its superior dialect classification performance, while IndicConformer is evaluated zero-shot to provide a cross-lingual transfer baseline.

3.5 Automatic Speech Recognition Architecture

3.5.1 Whisper

Whisper encoder uses multi-head self-attention to model long-range dependencies in speech representations:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \quad (2)$$

where Q, K, and V denote the query, key, and value matrices.

We fine-tuned Whisper-v3-large using Low-Rank Adaptation (LoRA). This reduces memory usage and catastrophic forgetting. Specifically, LoRA modules are applied to the query, key, and value projection matrices, as well as to the two feed-forward transformation layers across all layers of the model.

3.5.2 IndicConformer

The Conformer architecture uses convolutional neural networks and self-attention mechanisms to effectively capture both local acoustic patterns and global contextual dependencies in the speech signals. IndicConformer follows the original Conformer design (Gulati et al., 2020). Here, the Conformer model is evaluated in a zero-shot fashion.

4 Experiments

4.1 Experimental Setup

All experiments were conducted on a server with two NVIDIA T4 GPUs. For the DC task (Speech Foundation Models), we used a per-device batch size of 2 with gradient accumulation over 4 steps (effective batch size 8). For ASR, Whisper-v3-large was fine-tuned with a batch size of 1 and 4 accumulation steps (effective batch size 4). We used the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1\times10^{-8}$). The learning rate was set to 1×10^{-4} for DC and 3×10^{-5} for ASR, with cosine decay and 0.1% warmup.

4.2 Datasets

The dataset comprises Tamil speech recordings from four major regional dialects — Southern, Northern, Western, and Central — supporting both Dialect Classification and Automatic Speech Recognition (ASR). We also used the IndicVoices dataset (Javed et al., 2024) to improve ASR.

4.3 Performance Metrics

For the DC task we report Accuracy, Precision, Recall, F1, and their macro-averaged variants:

$$\text{Acc.} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

$$\text{Prec.} = \frac{TP}{TP + FP}, \qquad \text{Rec.} = \frac{TP}{TP + FN} \quad (4)$$

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

$$\text{Macro-Metric} = \frac{1}{K}\sum_{k=1}^{K} \text{Metric}_k \quad (6)$$

Table 2: Dialect Classification Results

Model	Acc	Prec	Rec	Macro F1
Whisper	**0.64**	**0.49**	**0.52**	**0.46**
CLDNN	0.38	0.31	0.34	0.33
WavLM	0.37	0.30	0.32	0.30
Wav2Vec2	0.36	0.26	0.28	0.26

where K is the number of classes. For ASR, we report Word Error Rate (WER) and Character Error Rate (CER):

$$\text{WER/CER} = \frac{S + D + I}{N} \tag{7}$$

where S, D, and I denote substitutions, deletions, and insertions, and N is the total number of words (WER) or characters (CER).

5 Results and Analysis

Tables 2 and 3 show that Whisper achieves the best performance for both dialect classification and ASR due to its pretrained speech representations. Combining IndicVoices with dialect-specific data improves ASR robustness.

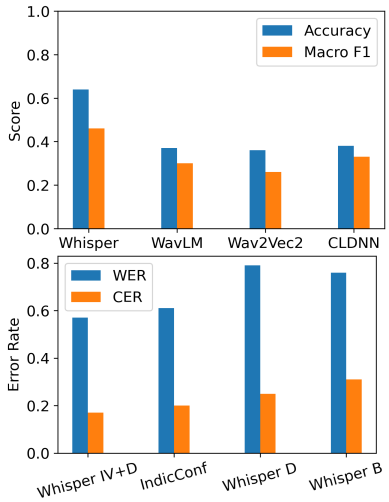


Figure 2: Dialect classification performance (top) and ASR error rates (bottom).

Whisper performs best on Southern dialects, moderately on Northern, and poorly on Central and Western due to limited training data.

6 Conclusion

In this work, we evaluated deep learning and speech foundation models for Tamil dialect classification and ASR. Whisper achieved the best performance with 0.46 Macro F1 and 0.57 WER.

Table 3: ASR Results

Model	WER	CER
Whisper (IndVoic.+Dia.)	**0.57**	**0.17**
Indic-Conformer	0.61	0.20
Whisper (Dia.)	0.79	0.25
Whisper (Base)	0.76	0.31

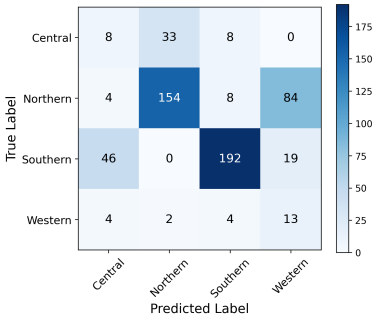


Figure 3: Confusion Matrix of Whisper for Dialect Classification.

Results show that combining general and dialect-specific speech data improves ASR robustness, while low-resource dialects remain challenging. As future work, we plan to explore a unified Whisper-based framework for both dialect classification and ASR by introducing dialect-specific tokens, enabling dialect-aware speech recognition.

References

B. Bharathi, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, S. Saranya, and S. Suhasini. 2026. Findings in Tamil Dialect Speech Recognition and Classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

B Bharathi, S Saranya, P Vijayalakshmi, and T Nagarajan. 2025. Multi-dialect speech corpus creation for enhancing tamil automatic speech recognition. *Circuits, Systems, and Signal Processing*, pages 1–19.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Long Zhou, Yanmin Qian, and 1 others. 2021. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *arXiv preprint arXiv:2110.13900*.

Pourdamghani Ghahremani and 1 others. 2015. Investigation of cnn□dnn and cnn□rnn architectures for speech recognition. In *Proceedings of Interspeech 2015*, pages 2019–2023.

Anmol Gulati, James Qin, Chung□Cheng Chiu, and 1 others. 2020. Conformer: Convolution□augmented transformer for speech recognition. In *Proceedings of Interspeech 2020*, pages 5036–5040.

Camille Harris, Chijioke Mgbahurike, Neha Kumar, and Diyi Yang. 2024. Modeling gender and dialect bias in automatic speech recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15166–15184, Miami, Florida, USA. Association for Computational Linguistics.

Nikhil Jakhar, Sudhanshu Srivastava, and Arun Baby. 2024. A Unified Approach to Multilingual Automatic Speech Recognition with Improved Language Identification for Indic Languages. In *Interspeech 2024*, pages 3949–3953.

Tahir Javed, Janki Nawale, Eldho George, Sakshi Joshi, Kaushal Bhogale, Deovrat Mehendale, Ishvinder Sethi, Aparna Ananthanarayanan, Hafsah Faquih, Pratiti Palit, Sneha Ravishankar, Saranya Sukumaran, Tripura Panchagnula, Sunjay Murali, Kunal Gandhi, Ambujavalli R, Manickam M, C Vaijayanthi, Krishnan Karunganni, and 2 others. 2024. IndicVoices: Towards building an inclusive multilingual speech dataset for Indian languages. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10740–10782, Bangkok, Thailand. Association for Computational Linguistics.

Shashidhar G. Koolagudi, Akash Bharadwaj, Y. V. Srinivasa Murthy, Nishaanth Reddy, and Priya Rao. 2017. Dravidian language classification from speech signal using spectral and prosodic features. *International Journal of Speech Technology*, 20(4):1005–1016.

Bo Li, Tara N. Sainath, Khe Chai Sim, Michiel Bacchiani, Eugene Weinstein, Patrick Nguyen, Zhifeng Chen, Yanghui Wu, and Kanishka Rao. 2018. Multi-dialect speech recognition with a single sequence-to-sequence model. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4749–4753.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre□training for speech recognition. *arXiv preprint arXiv:1904.05862*.

Shweta Sinha, Aruna Jain, and S. S. Agrawal. 2015. Fusion of multi-stream speech features for dialect classification. *CSI Transactions on ICT*, 2(4):243–252.

IndiLangTech@DravidianLangTech 2026: Hierarchical Modeling for Multi-Level Political Meme Classification

Saurabh Kumar, Vivekananda G, Sanasam Ranbir Singh, and Sukumar Nandi
Department of Computer Science and Engineering,
Indian Institute of Technology Guwahati, Guwahati-781039, India
{saurabh1003, g.vivekanada, ranbir, sukumar}@iitg.ac.in

Abstract

Political memes are a widely used form of digital political expression in linguistically diverse regions such as South India, where visual cues, textual overlays, and cultural symbolism convey complex political narratives. The Shared Task on Multi-Level Political Meme Classification at DravidianLangTech 2026 introduces a hierarchical setting requiring stance identification (Support vs. Troll) and target-type prediction (Individual vs. Party) for Tamil and Malayalam memes. We propose a two-stage hierarchical framework based on the Gemma 3 4B Instruction model. Instead of jointly predicting both levels, two specialized models are fine-tuned: the first predicts meme stance, and its output conditions the second model for target identification, explicitly modeling the dependency between the meme content, the predicted stance, and the target type. Using LoRA-based parameter-efficient instruction tuning, our approach achieves an average F1-scores of 0.8029 for Tamil and 0.6950 for Malayalam across both levels, ranking **1st** in Tamil and **4th** in Malayalam.

1 Introduction

Memes have become a prevalent form of communication on social media platforms, enabling users to express opinions, emotions, and attitudes on various social and political issues. Typically memes combining images or videos with short text overlays convey messages in a compact and highly shareable format (Atanasov et al., 2019; Kiela et al., 2021). While many memes are created purely for entertainment, others are used to shape public opinions or to target individuals, organizations, or political groups. By combining visual indicators, textual cues, satire, and cultural references, memes can communicate complex political narratives in a concise manner. Particularly in political discourses, memes often comprise of symbolic imagery, party iconography, and culturally grounded humor to express support, criticism, or ideological positions. Such multimodal and context-dependent memes make the detection and the analysis of memes challenging (Farabi et al., 2024).

(a) Support for Individual (b) Support for Party

(c) Troll against Individual (d) Troll against Party

Figure 1: Multi-Level Political Meme Sample.

To advance research on political memes detection in Dravidian languages, DravidianLangTech@ACL 2026[1] (Rajiakodi et al., 2026) introduces a dataset and a Shared Task on Multi-Level Political Meme Classification. The task focuses on hierarchical two-level classification for Tamil and Malayalam political memes. Sample memes from the dataset are illustrated in Figure 1. The task is to identify (i) whether a meme expresses **Support** or **Troll** (Level 1), and (ii) whether the meme targets an **Individual** or a **Party** (Level 2).

In response to this task, we propose a hierarchical two-stage framework built on the Gemma 3

[1] https://www.codabench.org/competitions/11325/

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 273–278
July 3, 2026 ©2026 Association for Computational Linguistics

4B Instruction model. The first stage predicts the stance of the meme, and its output is then used as contextual input for the second stage to identify the target type. This design explicitly models the dependency $P(y_2 \mid M, y_1)$, helping to reduce cross-level inconsistencies. Our approach achieves an average F1-scores of 0.8029 for Tamil and 0.6950 for Malayalam across both levels, ranking **1st** in Tamil and **4th** in Malayalam.

2 Related Work

Understanding political memes in Dravidian languages requires integrating insights from multimodal analysis, sarcasm detection, hierarchical classification, and Multimodal Large Language Models (MLLMs).

Early research on online toxicity and troll detection primarily relied on unimodal text-based natural language processing techniques. These studies formulated troll detection as a text classification task using sentiment analysis, n-gram features, and word embeddings to identify malicious or disruptive content in forums and social media posts (Fortuna and Nunes, 2018; Monakhov, 2020). While effective for standard textual data, such approaches often fail when applied to memes, where textual content may appear benign but becomes offensive or sarcastic when interpreted alongside visual context.

To overcome these limitations, subsequent work shifted toward multimodal approaches. The introduction of the Hateful Memes Challenge (Cai et al., 2019; Kiela et al., 2020) established an important benchmark for studying multimodal toxicity, demonstrating that benign visual and textual elements can combine to produce harmful meanings. Later studies explored cross-modal attention mechanisms and contrastive learning to capture complex interactions between images and embedded text in memes (Burbi et al., 2023; Arya and Bagwari, 2024).

Building on these advances, research extended to culturally rich and linguistically diverse Dravidian languages. Shared tasks at *DravidianLangTech* introduced datasets for Tamil and Malayalam troll meme classification (Suryawanshi et al., 2020; Suryawanshi and Chakravarthi, 2021; Premjith et al., 2022). Many approaches adopted early fusion strategies, combining Convolutional Neural Networks (CNNs) such as VGG16, VGG19, ResNet, and EfficientNet for visual feature extraction with transformer-based language models like mBERT, MuRIL, ALBERT, and XLNet to process multilingual or code-mixed text (Nandi et al., 2022; Das et al., 2022; Hariprasad et al., 2022; Krishna and Kumar, 2022; Hasan et al., 2022). Although these models improved meme classification performance, they generally treated the problem as a flat classification task without modeling dependencies between labels.

Recent advances in MLLMs have further improved performance on vision-language reasoning and meme understanding tasks (Liu et al., 2024; Team et al., 2024). However, most existing approaches still predict labels independently. In political meme analysis, this assumption can be limiting because the stance expressed in a meme often constrains its target. For example, a supportive meme may promote a political party or leader, whereas a trolling meme may criticize or attack a specific individual. Modeling such hierarchical relationships can therefore improve prediction consistency, as demonstrated in structured classification literature (Mohammad et al., 2016; Pramanick et al., 2021; Zhou et al., 2020; Chen et al., 2021).

Language	Level	Label	Train	Test
Tamil	1	Troll	691	175
		Support	112	26
	2	TAI	547	137
		TAP	146	37
		SFI	86	22
		SFP	24	5
Malayalam	1	Troll	477	96
		Support	23	4
	2	TAI	315	50
		TAP	110	31
		Inter	53	15
		SFI	12	4
		SFP	10	-

Table 1: Dataset statistics for Tamil and Malayalam meme images. Level-1 contains two labels: **Troll** and **Support**. Level-2 further categorizes these into Troll against Individual (**TAI**), Troll against Party (**TAP**), Support for Individual (**SFI**), Support for Party (**SFP**), and Intersection (**Inter**).

3 System Overview and Experimental Setups

This section describes the dataset, the task formulation, and the proposed hierarchical framework. We also outline the instruction-tuning strategy and

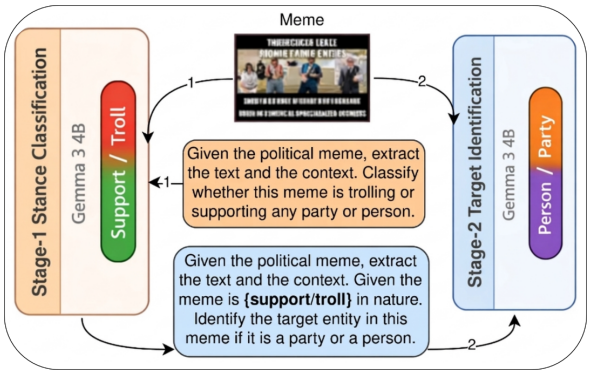


Figure 2: Two stage Hierarchical Framework for Multi-level Meme Classification

Component	Value
Base Model	gemma-3-4b-it
Fine-tuning	LoRA
LoRA Rank (r)	8
LoRA Alpha	16
Optimizer	AdamW
Learning Rate	5×10^{-5}
Scheduler	Cosine
Epochs	3
Per-device Batch Size	2
Gradient Accumulation	8
Effective Batch Size	16
Max Sequence Length	2048
Max Gradient Norm	1.0

Table 2: Training configuration for both models.

training configuration used for fine-tuning the multimodal model.

3.1 Dataset

We use the dataset released for the Shared Task on Multi-Level Political Meme Classification at DravidianLangTech 2026. The dataset consists of political meme images in two Dravidian languages: Tamil and Malayalam, annotated at two hierarchical levels.

Level 1 (Stance): Determines whether a meme expresses *Support* or *Troll* towards a political entity.

Level 2 (Target Identification): Identifies the type of political target referenced in the meme, such as an individual leader or a political party.

The Tamil dataset contains 1,003 memes (802 train, 201 test), while the Malayalam dataset contains 600 memes (500 train, 100 test). Detailed label distributions are presented in Table 1.

3.2 Hierarchical Two-Stage Framework

We implement a hierarchical two-stage framework based on the **Gemma 3 4B Instruction** model (Team et al., 2025), as illustrated in Figure 2. Gemma 3 is a multimodal large language model with strong vision–language understanding capabilities, making it suitable for analyzing memes that combine visual and textual cues. It employs a SigLIP vision encoder (Zhai et al., 2023) to process image inputs.

Instead of jointly predicting both levels, we fine-tune two specialized models that share the same pretrained backbone model but are optimized independently using instruction tuning. The first model (**Stage–1**) predicts the stance label, determining whether a meme expresses *support* or *troll*. The second model (**Stage–2**) predicts the target type, identifying whether the meme refers to an *individual* or a *party*. During the inference, the prediction from the Stage–1 model is passed as contextual input to the Stage–2 model, enabling to capture the hierarchical structure at the time of prediction.

Training instances follow an instruction-based format where the image meme and a task-specific prompt are provided as input, and the model generates a single-label response. We use the prompts described below to fine-tune the models for each stage. Unlike Tamil in the Malayalam dataset, an additional Level–2 target category namely *Intersection*, is present. To accommodate this label, the Stage–2 prompt is extended to include *Intersection* as an additional target entity.

Stage–1 Prompt

> `<image>`
> Given the political meme, analyze the visual and textual content. Classify whether the meme expresses trolling or support toward a political entity.
> Respond with only one label: “troll” or “support”.

Stage–2 Prompt

> `<image>`
> Given the political meme, analyze the visual and textual content. The meme expresses `[support/troll]`. Identify whether the target entity is a party or a person.
> Respond with only one label: “party” or “person”.

During training of the Stage–2 model, the gold Level–1 label is used as conditioning context, whereas during inference the predicted output from the Stage–1 model is supplied.

3.3 Training Configuration

We use the `Gemma 3 4B` instruction-tuned model (Team et al., 2025) as the backbone for both the stages. Both the models are fine-tuned using supervised instruction tuning with cross-entropy loss

Lang	Setups	Level-1				Level-2				Avg F1
		Acc	Prec	Recall	F1	Acc	Prec	Recall	F1	
ta	Independent	0.9204	0.9147	0.9204	0.9159	0.7065	0.6754	0.7065	0.6885	0.8022
	Hierarchical	0.9204	0.9147	0.9204	0.9159	0.6915	0.6930	0.6915	0.6898	**0.8029**
ml	Independent	0.9400	0.9208	0.9400	0.9303	0.5000	0.4003	0.5000	0.4243	0.6773
	Hierarchical	0.9400	0.9208	0.9400	0.9303	0.4800	0.4462	0.4800	0.4596	**0.6950**

Table 3: Performance of different training setups in terms of Accuracy (**Acc**), Precision (**Prec**), Recall (**Rec**), and F1-score (**F1**) for each level on Tamil (**ta**) and Malayalam (**ml**). **Avg F1** denotes the average F1-score across both.

and *LoRA-based parameter-efficient fine-tuning*. Low-rank adapters are inserted into the backbone while the vision encoder and multimodal projection layers remain frozen, reducing the number of trainable parameters while preserving pretrained multimodal representations.

Training is conducted using the LLaMA-Factory framework[2] for efficient multimodal instruction tuning. Mixed-precision training (`bf16`) is employed together with the AdamW optimizer, cosine learning rate scheduling, and gradient accumulation to improve computational efficiency. The detailed training configuration is summarized in Table 2. All implementation details, training scripts, and prompts used in this work are publicly available in the GitHub repository[3].

4 Results and Evaluation

We evaluate our proposed hierarchical framework on the official test set of the shared task. To analyze the impact of hierarchical modeling, we compare it with an **Independent** training setup using the same `Gemma-3-4B-it` backbone.

Independent: Two separate models are instruction fine-tuned independently for each task level: one for Level-1 stance classification and another for Level-2 target identification.

Hierarchical: Our proposed method follows a two-stage framework where Model–1 predicts the stance label (support or troll), and the predicted stance is used as contextual input for Model–2 to identify the target type (person or party). To ensure a fair comparison, we reuse the Level-1 model from the Independent setup and only train the Stage–2 model to incorporate the stance information predicted by Stage–1.

Table 3 presents the evaluation results for both Tamil and Malayalam datasets. From the results, we observe that Level-1 stance classification achieves consistently high performance across both languages. For Tamil we achieve an F1-score of 0.9159, while for Malayalam the models achieve an F1-score of 0.9303. This indicates that determining whether a meme expresses support or trolling is relatively easier for the model compared to identifying the specific target.

In contrast, Level-2 target identification is more challenging. For Tamil, the Independent setup achieves an F1-score of 0.6885, while the Hierarchical model slightly improves the performance to 0.6898. Although the improvement is modest, the hierarchical setup increases the precision from 0.6754 to 0.6930, suggesting that incorporating stance information helps the model make more accurate target predictions. A similar trend is observed for the Malayalam dataset. While the Independent model achieves an F1-score of 0.4243 for Level-2, the Hierarchical framework improves the F1-score to 0.4596 along with an increase in precision from 0.4003 to 0.4462. This indicates that conditioning the target prediction on the predicted stance helps the model better identify the relevant political entity.

4.1 Error Analysis

We perform a detailed error analysis of misclassified samples across both levels of the task. For Level-1 (stance classification), the model errors are primarily attributed to the presence of sarcastic or ironic textual content, as illustrated in Figure 3(a). Such cases often convey implicit meaning that contradicts the literal interpretation of the text. Additionally, memes containing ambiguous or mixed signals—where both supportive and trolling elements are present within the same instance—pose significant challenges to the model. Examples of such cases are shown in Figure 3(b) and (c), where

[2] https://github.com/hiyouga/LlamaFactory

[3] https://github.com/saurabhdbz/IndiLangTech-2026

(a) Misclassified as Troll (b) Misclassified as Troll (c) Misclassified as Support (d) Misclassified as Troll Against Party

Figure 3: Examples of misclassified memes highlighting challenges such as sarcasm and ambiguous intent.

the model struggles to resolve conflicting cues and assigns incorrect labels.

At Level-2 (target identification), errors mainly arise when both a political party and an individual are referenced within the same meme. In such cases, the model finds it difficult to determine the primary target entity, leading to incorrect predictions. An example of this ambiguity is shown in Figure 3(d), where the meme references both the individual (*Vijay*) and his political party (*TVK*). The model incorrectly predicts the target as a party and classifies the meme as trolling against the party, highlighting its difficulty in distinguishing the dominant target in multi-entity contexts.

5 Conclusion

This paper presented a hierarchical two-stage framework for multi-level political meme classification in Tamil and Malayalam using the Gemma 3 4B Instruction multimodal model. The proposed approach decomposes the task into stance detection and target identification, where the prediction from the first stage guides the second stage, enabling structured modeling of dependencies between the two levels. Experimental results demonstrate the effectiveness of this framework, achieving average F1-scores of 0.8029 for Tamil and 0.6950 for Malayalam, and securing the 1st rank for Tamil and 4th for Malayalam in the shared task.

Our analysis further reveals key challenges in political meme understanding, including sarcasm, implicit meaning, and ambiguity arising from mixed signals or multiple referenced entities. In particular, the model struggles when both a political figure and a party are present, highlighting limitations in resolving dominant targets. These findings suggest that incorporating improved multimodal reasoning, entity grounding, and context-aware interpretation could further enhance performance in such complex settings.

References

Gaurav Arya and Ashutosh Bagwari. 2024. Multimodal hate speech detection in memes using contrastive language-image pre-training. *IEEE Access*, 12:22359–22375.

Atanas Atanasov, Gianmarco De Francisci Morales, and Preslav Nakov. 2019. Predicting the role of political trolls in social media. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 1023–1034.

Giovanni Burbi, Alberto Baldrati, Lorenzo Agnolucci, Marco Bertini, and Alberto Del Bimbo. 2023. Mapping memes to words for multimodal hateful meme classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2014–2024.

Yitao Cai, Huiyu Cai, and Xiaojun Wan. 2019. Multi-modal sarcasm detection in twitter with hierarchical fusion model. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 2506–2515.

Haibin Chen, Qianli Ma, Zhenxi Lin, and Jiangong Jiang. 2021. Hierarchy-aware label semantics matching network for hierarchical text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 4370–4379.

Mithun Das, Somnath Banerjee, and Animesh Mukherjee. 2022. hate-alert@ dravidianlangtech-acl2022: Ensembling multi-modalities for tamil trollmeme classification. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 51–57.

Shafkat Farabi, Tharindu Ranasinghe, Diptesh Kanojia, Yu Kong, and Marcos Zampieri. 2024. A survey of multimodal sarcasm detection. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 8020–8028. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30.

Shruthi Hariprasad, Sarika Esackimuthu, Saritha Madhavan, Rajalakshmi Sivanaiah, and 1 others. 2022. Ssn_mlrg1@ dravidianlangtech-acl2022: Troll meme classification in tamil using transformer models. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 132–137.

Md Hasan, Nusratul Jannat, Eftekhar Hossain, Omar Sharif, and Mohammed Moshiul Hoque. 2022. Cuetnlp@ dravidianlangtech-acl2022: Investigating deep learning techniques to detect multimodal troll memes. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 170–176.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Casey A Fitzpatrick, Peter Bull, Greg Lipstein, Tony Nelli, Ron Zhu, and 1 others. 2021. The hateful memes challenge: Competition report. In *NeurIPS 2020 Competition and Demonstration Track*, pages 344–360. PMLR.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Advances in Neural Information Processing Systems*, volume 33, pages 2611–2624.

Achyuta Krishna and Mithun Kumar. 2022. Troll meme classification using feature extraction and transformer models. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. In *Advances in neural information processing systems*, volume 36.

Saif M Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41.

Sergei Monakhov. 2020. Understanding troll writing as a linguistic phenomenon. In *Proceedings of SAI Intelligent Systems Conference*, pages 315–334. Springer.

Rabindra Nath Nandi, Firoj Alam, and Preslav Nakov. 2022. Teamx@ dravidianlangtech-acl2022: A comparative analysis for troll-based meme classification. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 79–85.

Shraman Pramanick, Shivam Sharma, Dimitar Dimitrov, Preslav Nakov, Pawan Goyal, Animesh Bhattacharya, Md. Arid Hasan, Niloy Bhattacharya, and Sourya Das. 2021. Detecting harmful memes and their targets. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2783–2796. Association for Computational Linguistics.

B Premjith, Bharathi Raja Chakravarthi, Malliga Subramanian, B Bharathi, Soman Kp, V Dhanalakshmi, K Sreelakshmi, Arunaggiri Pandian, and Prasanna Kumaresan. 2022. Findings of the shared task on multimodal sentiment analysis and troll meme classification in dravidian languages. In *Proceedings of the second workshop on speech and language technologies for Dravidian languages*, pages 254–260.

Saranya Rajiakodi, Shunmuga Priya Muthusamy Chinnan, Premjith B, Subalalitha CN, Rahul Ponnusamy, Anshid K A, Bhuvaneshwari Sivagnanam, Jananayagan V, Bharathi Raja Chakravarthi, Ragavan N, and Santhini P. 2026. Overview of the shared task on multilevel political meme classification in tamil and malayalam. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Shardul Suryawanshi and Bharathi Raja Chakravarthi. 2021. Findings of the shared task on troll meme classification in Tamil. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 126–132.

Shardul Suryawanshi, Bharathi Raja Chakravarthi, Mihael Arcan, and Paul Buitelaar. 2020. Multimodal meme dataset (multioff) for identifying offensive content in image and text. In *Proceedings of the second workshop on trolling, aggression and cyberbullying*, pages 32–41.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Gemma Team, Thomas Mesnard, Cassidy Hardin, and 1 others. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986.

Jie Zhou, Chunping Ma, Dingkun Long, and 1 others. 2020. Hierarchy-aware global model for hierarchical text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1106–1117.

JerinWarriors@DravidianLangTech 2026: A Two-Stream Cross-Attention Approach for Prompt Recovery in Telugu

Savith Arivudainambi
IISER Kolkata, West Bengal, India
sa24ms140@iiserkol.ac.in

Wordson Robert
IISER Kolkata, West Bengal, India
wr24ms190@iiserkol.ac.in

C. Jerin Mahibha
Meenakshi Sundararajan Engineering College, Chennai, Tamil Nadu, India
jerinmahibha@msec.edu.in

Shrey Patnaik
IISER Kolkata, West Bengal, India
sp24ms174@iiserkol.ac.in

Abstract

Identifying the structure of detailed sentences which show glimpses of various annotation cues, in a low resource language that is morphological rich like Telugu is a challenge. Standard baseline architectures like Multi Layer Perceptrons (MLP) struggle with low resource languages. This paper details our proposed solution for the Telugu Prompt-Style Recovery Shared Task at DravidianLangTech @ ACL 2026. We propose a Two-Stream Cross-Attention architecture that uses a shared MuRIL encoder to calculate the relationship between an original transcript and its style-shifted counterpart, helping the MLP to distinguish the styles better and catch the differences better. Through experimentation we have found out that this proposed model handles the signal dilution of the individual labels better than the rest. Our best-performing system achieved a Macro F1-score of 0.2588 on the test set, securing 2nd place out of 13 teams. We have concluded that the local transformation is the main driver for the style recovery in this task. For reproducibility, we release our implementation and experimental setup on GitHub.[1]

1 Introduction

Recent advances in Natural Language Processing (NLP) have significantly improved the ability to analyze social media text in low-resource languages. In the context of Dravidian languages, several studies have explored tasks such as sarcasm detection, sentiment analysis, and offensive language identification using machine learning and transformer-based approaches. For example, prior work has investigated sarcasm detection in Dravidian languages using transformer architectures, demonstrating the effectiveness of contextual language models in capturing subtle linguistic cues in social media text (Madhumitha et al., 2023). Similarly, studies on offensive language identification have applied both traditional machine learning and deep learning techniques to detect abusive or harmful content in regional language datasets (Mahibha et al., 2021b).

Other research has explored sentiment analysis in multilingual and cross-lingual settings, showing that cross-lingual word embedding models can effectively transfer semantic knowledge across languages with limited annotated data (Mahibha et al., 2021a). Additionally, sarcasm detection from Dravidian language text has been examined as an important task for understanding nuanced expressions in online communication (Mahibha et al., 2024). These studies highlight the growing interest in applying advanced NLP techniques to Dravidian languages and demonstrate the need for models that can effectively capture linguistic complexity in these morphologically rich languages.

Recognizing the intent and the style used in low resource Indian languages is a very under-focused area in NLP research. The regular use of sentiment analysis fails these languages because they are not trained on Dravidian languages. These context-rich sentences from morphologically diverse languages, especially with context, prove to be tough to classify into categories, namely Formal, Informal, Optimistic, Pessimistic, Humorous, Serious, Inspiring, Authoritative, and Persuasive.

We ran several models, with several different parameters, and we tried a myriad of different combinations of techniques. We found that almost 65 percent of stylistic changes in the paragraphs happen within the first 256 words. We found out that the usual techniques, such as the hierarchical models, fail spectacularly at aggregating their signals, resulting in an F1 score of 2 to 5 percent. This proves that extensive signal dilution happens over these long sentences. The model learned best when it was learning from the semantic difference between the original text and the rewritten version, indicating that the sentences are not captured well

[1] https://github.com/WordsonRobert/telugu-prompt-recovery-llm-acl-2026-shared-task

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 279–283
July 3, 2026 ©2026 Association for Computational Linguistics

by the tone of the original sentences alone, meaning the styling is not absolute.

Keeping all these in mind, we implemented a Two-Stream Cross-Attention Classifier built on the MuRIL (Multilingual Representations for Indian Languages) backbone. By allowing the style-changed text to compare with the original transcript, our model captures the delta in intent and framing as shown by the results that we have achieved in the shared task (Premjith et al., 2026).

2 Related Work

The field of Telugu Natural Language Processing has evolved significantly with the advent of transformer-based architectures. Dowlagar and Mamidi (2021) established foundational benchmarks by evaluating BERT and Multilingual BERT (mBERT) for hate speech detection in Telugu. Their research highlighted that while mBERT provides a robust starting point, the language's agglutinative nature requires specialized fine-tuning to capture subtle morphological shifts. Building upon this, Kakarla and Venkata (2025) focused on the complexities of code-mixed Telugu-English content. By utilizing models like TeluguHateBERT and Hindi-Abusive-MuRIL, they demonstrated that pre-training on linguistically similar Indian languages significantly improves the detection of aggressive intent compared to generic multilingual models.

In the broader context of Dravidian languages, Dave et al. (2021) presented a hybrid approach combining MuRIL embeddings with character N-grams. Their work in the DravidianLangTech workshop proved that MuRIL is superior for low-resource scripts like Telugu because it retains regional semantic nuances that are often "washed out" in globally trained models. The effectiveness of multilingual transformer representations for cross-lingual NLP tasks was earlier demonstrated by Devlin et al. (2019), whose Bidirectional Encoder Representations from Transformers (BERT) architecture laid the foundation for contextualized language modelling. Subsequent work on multilingual contextual models such as mBERT showed that shared multilingual representations can transfer knowledge across languages, including low-resource ones (Pires et al., 2019).

Furthermore, Arunachalam and Maheswari (2024) introduced ensemble transformer methods to address social media cleanup. Their study emphasized that custom preprocessing, such as aggressive stemming, is vital for Dravidian scripts to reduce vocabulary sparsity. The development of MuRIL, a multilingual transformer specifically designed for Indian languages, further improved performance on Indic language tasks by incorporating transliteration and parallel corpora during training (Khanuja et al., 2021). Finally, the theoretical shift toward dual-representation learning is supported by Duan et al. (2022), whose work on two-stream transformers showed that cross-attention fusion between original and modified text segments allows a model to adaptively extract "difference features" that single-stream encoders often ignore. These studies collectively justify the use of a MuRIL-based two-stream cross-attention model for stylistic and offensive speech classification in Telugu.

3 Data Preparation and Style Definitions

The dataset for this task consists of Telugu transcript excerpts paired with their style-shifted versions. To ensure model robustness and prevent label leakage, we performed several preprocessing steps prior to training.

3.1 Data Labeling

The task requires classification into nine distinct categories. We mapped these labels to a numerical index as follows:

ID	Stylistic Category
0	Authoritative
1	Formal
2	Humorous
3	Informal
4	Inspiring
5	Optimistic
6	Persuasive
7	Pessimistic
8	Serious

Table 1: Label mapping for the 9-way Telugu style classification.

3.2 Preprocessing

We utilized the `google/muril-base-cased` tokenizer to process the Telugu Unicode script. Based on our locality analysis, which indicated that over 60% of stylistic cues are concentrated at the beginning of the text, we applied a truncation strategy with a **maximum sequence length of 256 tokens**. In the test phase, where original transcripts were unavailable, the architecture transitioned to a

self-attention configuration by supplying the style-shifted text to both input branches. Although explicit semantic comparison between original and transformed transcripts could not be performed during inference, this design preserved architectural consistency across the feature extraction layers. The representations learned during cross-attention training continued to provide stylistically discriminative features, enabling effective generalization even in the absence of reference transcripts.

4 Methodology

Our proposed architecture, the **Two-Stream Cross-Attention Classifier**, is designed to capture the stylistic "delta" between an original Telugu transcript and its rewritten version. Unlike single-stream models that treat style as an absolute property of a single text, our approach models style as a relational transformation.

4.1 Formalism

Let $\mathcal{S}_{orig}$ and $\mathcal{S}_{chg}$ denote the input sequences for the original transcript and the style-shifted text, respectively. Both sequences are passed through a shared MuRIL encoder $\mathcal{E}$ to obtain dense hidden representations:

$$H_{orig} = \mathcal{E}(\mathcal{S}_{orig}), \quad H_{chg} = \mathcal{E}(\mathcal{S}_{chg}) \tag{1}$$

where $H \in \mathbb{R}^{L \times d}$ (with $L = 256$ and $d = 768$). To capture the specific stylistic shifts, we implement a Cross-Attention mechanism where the style-changed hidden states act as the Query (Q), while the original hidden states provide the Key (K) and Value (V):

$$Q = H_{chg}W^Q, \quad K = H_{orig}W^K, \quad V = H_{orig}W^V \tag{2}$$

To prevent signal dilution from padding tokens, we apply a Key Padding Mask $M \in \{0, -\infty\}^L$. The attended representation Z is computed as:

$$Z = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}} + M\right)V \tag{3}$$

Following the attention operation, we apply a residual connection and Layer Normalization (LN) to stabilize the enriched features:

$$H_{final} = \text{LN}(H_{chg} + \text{Dropout}(Z)) \tag{4}$$

The final stylistic classification $\hat{y}$ is derived from the pooled representation of the `[CLS]` token at index 0. Figure 1 shows our complete architecture.

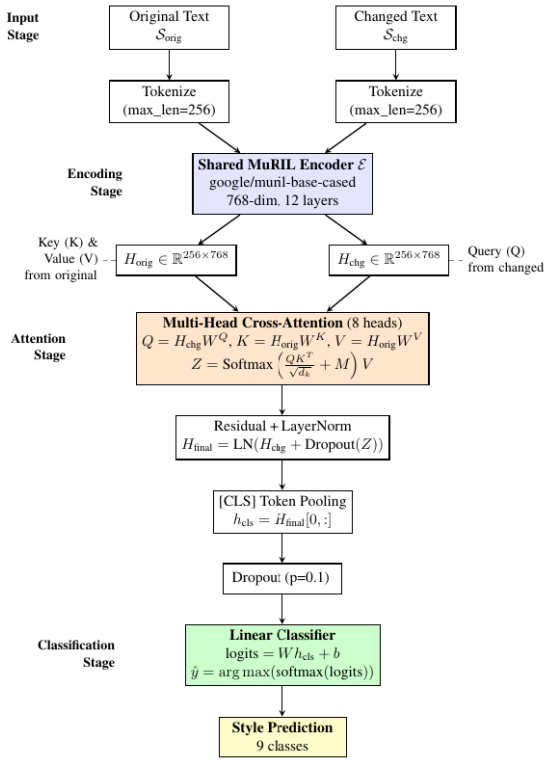


Figure 1: Two-Stream Cross-Attention architecture.

5 Results and Discussion

5.1 Performance Metrics

We evaluated the model over 5 epochs using the AdamW optimizer with a learning rate of 2×10^{-5}. The performance reached its peak at Epoch 3, as shown in Table 2.

Metric	Value
Total Samples	301
Correct	81 (26.9%)
Errors	220 (73.1%)
Accuracy	0.2691
F1 (Macro)	0.2588

Table 2: Test set performance.

5.2 Analysis of Findings

The Overfitting Pivot: We observed that while Training Loss continued to decline from 2.14 to 1.20, the Validation Loss began to rise after Epoch 3. This indicates that the model began to memorize specific vocabulary markers for the nine style labels rather than learning generalizable stylistic shifts.

Locality vs. Hierarchy: Our experiments with 256-token truncation were significantly more successful than hierarchical "chunking" methods. This

True → Predicted	Count	%
Serious → Formal	17	7.7%
Authoritative → Formal	12	5.5%
Humorous → Informal	11	5.0%
Inspiring → Persuasive	9	4.1%
Serious → Informal	9	4.1%
Optimistic → Persuasive	9	4.1%
Humorous → Formal	8	3.6%
Inspiring → Optimistic	8	3.6%
Other	167	75.9%
Total	**220**	**100.0%**

Table 3: Top misclassifications.

Style	Prec	Rec	F1
Authoritative	0.33	0.10	0.15
Formal	0.20	0.39	0.27
Humorous	0.35	0.26	0.30
Informal	0.27	0.43	0.33
Inspiring	0.33	0.11	0.16
Optimistic	0.34	0.36	0.35
Persuasive	0.24	0.43	0.31
Pessimistic	0.30	0.36	0.33
Serious	0.27	0.09	0.13
Macro	**0.29**	**0.28**	**0.26**

Table 4: Per-class metrics.

suggests that stylistic markers in Telugu are heavily localized; the dilution of signals in longer sequences (as seen in the 2-5% F1 scores of hierarchical models) is a major bottleneck for style recovery in low-resource languages.

5.3 Error Analysis

The confusion matrix represented in Figure 2 reveals three semantic clusters: Professional styles (Formal–Authoritative–Serious, 41 confusions), Casual styles (Informal–Humorous, 14 confusions), and Motivational styles (Inspiring–Optimistic–Persuasive, 26 confusions). The model systematically overpredicted Formal (75 predictions vs. 38 true) while underpredicting Serious (15 vs. 47 true). The count and percentage of misclassifications are represented in Table 3. The performance metrics such as precision, recall, and F1 score, associated with the different style categories are represented in Table 4. Best performers were Optimistic (F1=0.35) and Informal (F1=0.33) with clear lexical markers, while Serious (F1=0.13) and Authoritative (F1=0.15) struggled due to subtle tonal differences the 256-token window failed to capture.

6 Conclusion

This paper presented a Two-Stream Cross-Attention approach for the Telugu Prompt-Style Recovery task. Through cross-attention-based representation learning between original and style-shifted texts during training, the proposed system effectively captures stylistically discriminative semantic patterns in morphologically rich sentences. Experimental findings indicate that localized semantic transformations, particularly those occurring in the initial segments of the document, contribute significantly to style identification. Furthermore, the learned representations demonstrate strong generalization capability during inference even when the original reference transcript is unavailable. Future work will explore the use of RAG-based systems to recover missing transcripts in the test sets.

References

V Arunachalam and N Maheswari. 2024. Enhanced detection of hate speech in dravidian languages in social media using ensemble transformers. *Interdisciplinary Journal of Information, Knowledge, and Management*, 19:036.

Bhargav Dave, Shripad Bhat, and Prasenjit Majumder. 2021. Irnlp_daiict@ dravidianlangtech-eacl2021: offensive language identification in dravidian languages using tf-idf char n-grams and muril. In *Proceedings of the first workshop on speech and language technologies for Dravidian Languages*, pages 266–269.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

Suman Dowlagar and Radhika Mamidi. 2021. Hasocone@ fire-hasoc2020: Using bert and multilingual bert models for hate speech detection. *arXiv preprint arXiv:2101.09007*.

Lihua Duan, Qi You, Xinke Wu, and Jun Sun. 2022. Multilabel text classification algorithm based on fusion of two-stream transformer. *Electronics*, 11(14):2138.

Santhosh Kakarla and Gautama Shastry Bulusu Venkata. 2025. Code-mixed telugu-english hate speech detection. *arXiv preprint arXiv:2502.10632*.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, and Sebastian Ruder. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

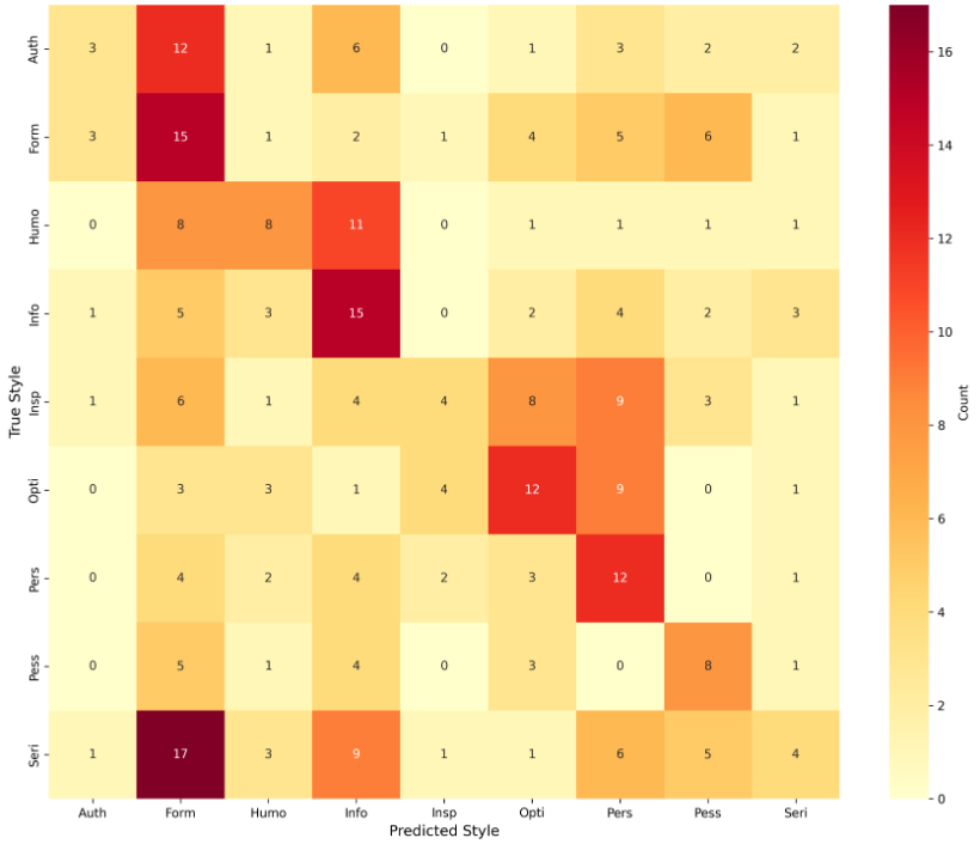


Figure 2: Confusion Matrix

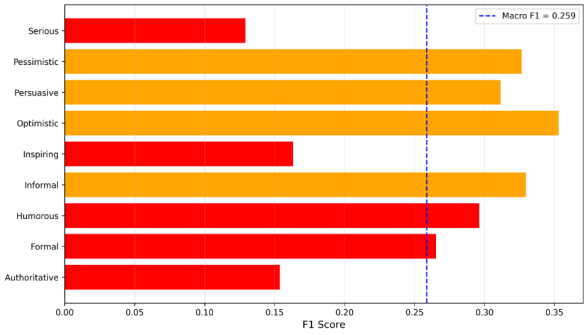


Figure 3: Per class F1-Score

M Madhumitha, Kunguma Akshatra M, J Tejashri, Mahibha C Jerin, and Durairaj Thenmozhi. 2023. Sarcasm detection in dravidian languages using transformer models. In *FIRE (Working Notes)*, pages 306–318.

C Jerin Mahibha, Kayalvizhi Sampath, and Durairaj Thenmozhi. 2021a. Sentiment analysis using cross lingual word embedding model. In *FIRE (Working Notes)*, pages 1094–1100.

C Jerin Mahibha, Kayalvizhi Sampath, Durairaj Thenmozhi, and S Arunima. 2021b. Offensive language identification using machine learning and deep learning techniques. In *FIRE (Working Notes)*, pages 705–713.

C Jerin Mahibha, Gersome Shimi, and Durairaj Thenmozhi. 2024. Sarcasm detection from dravidian language text. In *Forum of Information Retrieval and Evaluation FIRE-2024*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4996–5001.

B Premjith, G Jyothish Lal, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Thenmozhi Durairaj, Ratnavel Rajalakshmi, Rahul Ponnusamy, and Chinthala Bhuvanesh. 2026. Shared Task on Prompt Recovery for LLM in Telugu. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

KEC'S CODE CRAFTERS@DravidianLangTech 2026 : Abusive Tamil Text Detection Targeting Women on Social Media

Nethrasri S [1] **Nivetha V**[1] **Malliga Subramanian**[2]

[1]Department of CSE, Kongu Engineering College, Perundurai, Erode, India.
{nethrasris.24cse@kongu.edu, nivethav.24cse@kongu.edu}
mallisenthil.cse@kongu.edu}

Abstract

Social media is gaining popularity; however, it has become a platform where digital toxicity abounds. Linguistically diverse women find themselves highly vulnerable to cyberbullying. As a result, there is the necessity to develop a solution capable of tackling region-specific languages. Our paper takes steps in solving this issue by presenting a classification model for "Abusive Tamil Text Detection Targeting Women on Social Media" at the DravidianLangTech-2026 challenge. For developing our model, we used the dataset of 25,948 comments for training and 915 comments for testing. Our main task was the ability to classify content into two classes "Abusive" or "Non-Abusive" for YouTube videos. There were specific challenges associated with using the Tamil language, such as its complicated construction and the fact that the language is often mixed with English within the same sentence. In order to resolve these issues, we designed a pipeline for cleaning these informal scripts. Finally, we ran four traditional ML models: SVM, Logistic Regression, Random Forest, and Multinomial Naive Bayes using TF-IDF as a method of extracting features from texts. Our model achieved the accuracy and F1 scores of 0.86 using Logistic Regression.

1 Introduction

This has created the phenomenon of digital public space, where the level of interaction has touched new heights. Such digital violence is a serious issue that is manifested by silencing of female voices, inflicting psychological harm, and fostering gender disparity in both the physical world and the online world. Tamil is a Classical Dravidian language spoken by more than 70 million speakers globally. Detecting abuse in the Tamil language has been found to be a challenging exercise owing to the complicated nature of Tamil language and code-mixing between Tamil and English in the online world. In addition, the lack of high-quality dataset poses another problem. Thus, there is clearly a significant gap in the research area, which needs to be addressed properly. In order to address this gap, the current research paper attempts to do so by focusing on binary classification problem of Tamil comments found in YouTube video clips. Such a classification would include two categories: either Abusive or Non-Abusive comments with special emphasis on text against women. This research was conducted in relation to the task provided by the shared task at the DravidianLangTech-2026, which motivated this study. A joint effort was made to provide datasets which contain 25,948 comments in Tamil language from YouTube videos labeled for their abuse against women in the training set, and a dataset with 915 comments in the test set. The aim of this study is to develop and evaluate the effectiveness of classical machine learning algorithms for the current task. The specific steps taken in this study include the following: firstly, designing a custom preprocessing pipeline for Tamil text in social media, secondly, obtaining numeric representation through TF-IDF vectorization, and thirdly, evaluating the performance of four different classifiers including SVM, Logistic Regression, Random Forest, and Naive Bayes.

2 Literature Survey

Unfortunately, despite the presence of over 250M speakers of Dravidian language family, very little effort has been put into analyzing the problem of gender-targeted abuse with regards to the Dravidian language family. As a point of reference, the base model proposed for Dravidian language family in DravidianLangTech@ACL2023 scored a meager 0.65 F1 score using SVM and TF-IDF [8]. However, recent developments such as the presentation by Rahman et al. [1] at DravidianLangTech-2025 showed an increase in performance where they got

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 284–288
July 3, 2026 ©2026 Association for Computational Linguistics

0.86-0.87 score using XLM-BERT, while Subramanian et al. [2] managed 0.82-0.84 F1 using ensemble techniques. Further achievements include obtaining 0.84 using TF-IDF-BERT hybrid classifier by Harini et al. [3], while others such as Mohan et al. [4] propose multimodal approaches, among others [5]. In addition to this, papers dealing with the gender-targeted abuse subtask deal with issues related to gender-directed hate speech detection [6] (precision: 0.83, recall: 0.81 with Logistic Regression), misogyny detection using CNN [7] (Tamil, F1: 0.79), SVM-based Gender-directed Hate Speech Detection [8] (Tamil, F1: 0.81) and various transformer based methods [9]. The state-of-the-art for Tamil language abusive language detection is currently at 0.87 using ensemble methods [10].

3 Materials and Methods

3.1 Dataset Description

The dataset used in the current research is that which has been provided by the organizers of "Abusive Tamil Text Detection Targeting Women on Social Media" shared task of DravidianLangTech-2026. The dataset includes 26,863 comments made on YouTube videos in Tamil, out of which, the training set has 25,948 comments and the testing set has 915 comments. YouTube videos were chosen as the sources of the data because it offers an active and diverse Tamil speaking population along with an engaging comments section. A few YouTube videos were randomly chosen based on the criterion that those included comments targeting women. There are abusive as well as non-abusive categories of comments in the dataset, in similar numbers in order to have an efficient training of the model. All of the comments were manually labeled by fluent Tamil speakers and experts in the field.

Non-Abusive (Label 0) The non-abusive comments consist of any benign, positive, or neutral comments, discussions, and questions towards other users or compliments regarding women. It does not include abusive comments towards women. Abusive (Label 1) Comments that contain derogatory, sexist, threatening statements, name-calling towards women, and other terms targeting women (such as looks and sexual harassment), sexually abusive comments, and threatening remarks are considered abusive.

This work focuses on detecting abusive and non abusive text in Tamil language. In this particular work, we have added the feature to detect abuse toward women. It has an important difference when compared to typical English language data sets; as it focuses on a sensitive topic group and a low resource language.

Table 1: Dataset Statistics and Splits

Dataset Split	Total Comments	Language
Training Set	25,948	Tamil
Test Set	915	Tamil

3.2 Pre-processing and Feature Extraction

To get rid of the noise in the social media texts, the following preprocessing techniques were applied to the comments. Firstly, the use of regular expressions helped us to strip the URLs, mentions, hashtags, and special characters like $ % # etc. Secondly, the emojis found in the comments were converted to text-based emojis, preserving their context through the emojis python library. Thirdly, whitespace normalization reduced consecutive spaces, tabs, and new lines into one space character. Finally, all texts were converted into lowercase letters.

Next, the data was vectorized with TF-IDF vectorization technique. Term Frequency is a measure of how many times a word appears in a document. Inverse Document Frequency decreases the weights of commonly used words across the documents. Multiplying the two metrics provides an efficient representation of the document because it helps determine the importance of the term in the document corpus.

The choice of ngrams to be considered by TfidfVectorizer includes unigrams and bigrams, which capture individual words and phrases related to women-centric abuse (such as gender-specific slangs). The minimum document frequency is 2, removing uncommon words from the document corpus. Finally, we limit the feature size to 10000 dimensions.

4 Proposed Classifier

We have developed four classical machine learning models for abusive Tamil text detection for women on social media.

We picked four classic machine learning classifiers, each with its own approach to the problem. All came from scikit-learn. We split our training

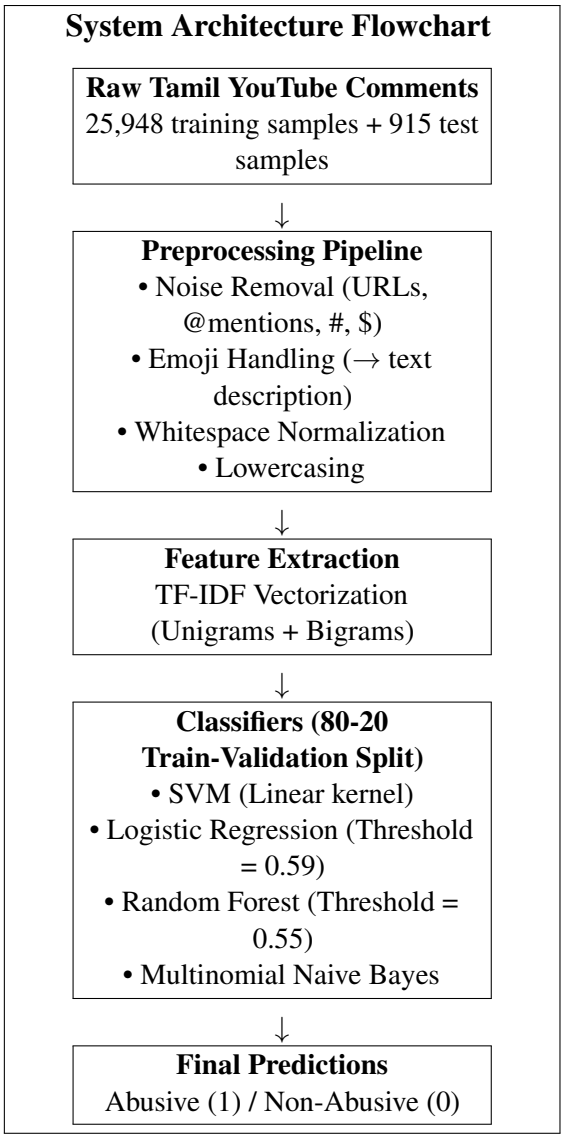


Figure 1: Complete system architecture for abusive Tamil text detection showing the preprocessing pipeline, feature extraction, and model training workflow with optimized parameters

data 80-20 into training and validation sets so we could tune hyperparameters and find the best thresholds for models that output probabilities.

4.1 Support Vector Machine (SVM)

We decided to implement linear SVM. Essentially, it looks for an optimal hyperplane that splits up abusive and non-abusive comments with maximal margin in that high-dimensional TF-IDF vector space. It is known to perform well with sparse, high-dimensional datasets such as text data. The default hyperparameters of LinearSVC was used here.

4.2 Logistic Regression (LR)

In Logistic Regression, probability of a comment being abusive is calculated by feeding an input of weighted features into what is called a sigmoid or a logistic function. The good thing about LR algorithm is that it is an interpretable model, meaning one could see which words contribute how to the final probability output. Moreover, the probabilities provided by LR are accurate. Some threshold tuning was performed for LR. By default, comments having probability of class ¿ 0.5 are classified as abusive. However, it turns out that we can tune the threshold to have a more accurate model. Our searches revealed that an optimal threshold is 0.59 on our validation dataset.

4.3 Random Forest (RF)

It is a machine learning algorithm that builds a large number of decision trees during training and outputs the class that is the most common among the classes output by individual trees. RF is able to model the complex nonlinear nature of the data set that linear classifiers cannot model. Similarly, like logistic regression, it can also predict probabilities. Key hyperparameters that will influence the prediction result of LR were selected: number of estimators (50, 100, 200) and maximum depth (10, 20, or None) using a grid search, and the optimal number of estimators (100) and unlimited depth were selected that provided the best validation accuracy with probability 0.55.

4.4 Naive Bayes (NB)

It is a probabilistic classifier based on the Bayes' theorem with the assumption that all predictors are independent of each other. That seems too simplistic, and realistically not possible because there are clear correlations between words in a sentence. Interestingly, Naive Bayes performs impressively well when predicting text-related problems such as frequency counts or TF-IDF features. Also, it is computationally efficient in terms of both training and predictions. The MultinomialNB method was utilized for features that are distributed according to a multinomial distribution.

5 Results

5.1 Performance Analysis

Logistic Regression had the best accuracy and F1 score of 0.86 which indicates the linear relationship between the TF-IDF features and the abusive

messages. SVM came second just like logistic regression having a F1 score of 0.84 and indicating that SVM works effectively on high-dimensional features. Random forest had low recall of 0.75 for the abusive label indicating that the model was overfitting the data. Naive bayes had the least F1 score of 0.80 since the features are not independent.

5.2 Why Logistic Regression Performed Best

In all the models analyzed, Logistic Regression gave the most optimal results with an accuracy and F1 score of 0.86 since TF-IDF features generate a sparse and high dimensional feature space which is highly appropriate for linear classifiers. The model could learn discriminative abusive terms while still generalizing well. Additionally, Logistic Regression generated calibrated probabilities which made it possible to optimize the threshold value to 0.59, thereby achieving a balance between precision and recall. SVM gave competitive results with an F1 score of 0.84, proving the efficiency of linear models when working with high dimensional text data. Random Forest had lower recall compared to the rest of the models as it was prone to overfitting with sparse lexical features. Naive Bayes yielded poor results since it could not model the context dependency of abusive terms in the Tamil social media text dataset.

5.3 Error Analysis

While showing good performance, the models had issues recognizing the abusive terms when they were indirect or context-based. False negatives were seen in cases where sarcasm was involved in the comment or the abusive sentiment was not explicitly mentioned but was implied in the comment. In many cases of false positives, comments with slang terms and emotional expressions, which were used in non-abusive settings, were wrongly labeled. The transliteration and code-mixing of Tamil-English comments made them harder to recognize due to inconsistencies in features caused by the use of phonetic spellings.

Table 2: Comparative Performance of All Models

Model	Prec%	Rec%	F1%	Acc%
LR+TF-IDF	86	86	86	86.0
SVM+TF-IDF	84	84	84	84.0
RF+TF-IDF	83	83	83	83.0
NB+TF-IDF	80	80	80	80.0

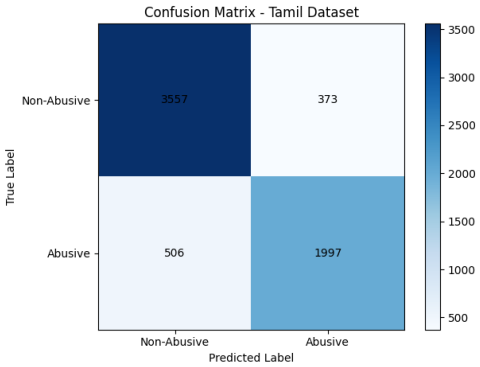


Figure 2: Confusion Matrix - Logistic Regression Model (Tamil Dataset)

6 Conclusion

This research work presented our approach towards the DravidianLangTech-2026 Shared Task for detecting abusive comments against women in the Tamil language. Our team designed four classical machine learning models utilizing TF-IDF features. Our proposed data preprocessing strategy proved to be highly effective for noisy Tamil social media data containing emojis, transliteration, code-switching text, and special characters.

From experimental analysis, Logistic Regression with parameter tuning turned out to be the best performing model among others like SVM, Random Forest, and Naive Bayes with an F1-score and accuracy of 0.86. The classifiers were still bound by their inability to deal with sarcasm, implied harassment and heavily mixed code language. Transliterated spellings of Tamil language and casual writing styles used on social media networks would further lower the stability of feature representations. Lastly, the current approach limited itself to textual data only, ignoring possible use of multimodal features, such as pictures and user profiles.

Future directions will include investigating transformer-based architectures like MuRIL and XLM-RoBERTa, along with ensemble modeling, contextual embeddings, and data augmentation approaches to enhance the effectiveness of semantic comprehension and abusive Tamil text identification tasks.

References

[1] Rahman, M.M., Dhar, S., Hasan, M.M., & Murad, H. (2025). MSM_CUET@DravidianLangTech 2025: XLM-BERT and MuRIL Based Transformer Models for Detection of Abusive Tamil and Malayalam Text Targeting Women on Social Media. In *Proceedings*

of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages (pp. 243-247). Association for Computational Linguistics.

[2] Subramanian, M., Shanmugavadivel, K., Indhuja, V.S., Kowshik, P., & Jayasurya, S. (2025). KECEmpower@DravidianLangTech 2025: Abusive Tamil and Malayalam Text targeting Women on Social Media. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages* (pp. 178-181). Association for Computational Linguistics.

[3] Harini, B.N.S., Meghana, K.V., Supriya, K., Samiksha, T., & Premjith, B. (2025). HTMS@DravidianLangTech 2025: Fusing TF-IDF and BERT with Dimensionality Reduction for Abusive Language Detection in Tamil and Malayalam. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages* (pp. 152-156). Association for Computational Linguistics.

[4] Mohan, J., Mekapati, S.R., Premjith, B., Jyothish Lal, G., & Chakravarthi, B.R. (2025). A Multimodal Approach for Hate and Offensive Content Detection in Tamil: From Corpus Creation to Model Development. *ACM Transactions on Asian and Low-Resource Language Information Processing*. DOI: 10.1145/3712260

[5] G, D., et al. (2025). Abusive Text Detection Targeting Women in Tamil and Malayalam using Machine Learning Classifiers. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages* (pp. XXX-XXX). Association for Computational Linguistics.

[6] Bade, G.Y., et al. (2025). GS_DravidianLangTech@2025: Women Targeted Abusive Texts Detection on Social Media. *arXiv preprint*, arXiv:2504.02863. https://arxiv.org/abs/2504.02863

[7] Shanmugavadivel, K., Subramanian, M., Pooja Sree, M., Palanimurugan, V., & Roshini Priya, K. (2025). InnovationEngineers@DravidianLangTech 2025: Enhanced CNN Models for Detecting Misogyny in Tamil Memes Using Image and Text Classification. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

[8] Vaidyanathan, V.K., Srihari, V.K., & Durairaj, T. (2025). NLP_goats@DravidianLangTech 2025: Towards Safer Social Media: Detecting Abusive Language Directed at Women in Dravidian Languages. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

[9] Thirumoorthy, S., Durairaj, T., & Rajalakshmi, R. (2025). Hydrangea@DravidianLanTech2025: Abusive language Identification from Tamil and Malayalam Text using Transformer Models. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

[10] Chakravarthi, B.R., Priyadharshini, R., Madasamy, A.K., Thavareesan, S., Sherly, E., Rajiakodi, S., Palani, B., Subramanian, M., Cn, S., & Chinnappa, D. (Eds.). (2025). *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Albuquerque, New Mexico: Association for Computational Linguistics.

code Availability: The implementation of our proposed model is publicly available at: https://github.com/NivethaVallarasu/dravidian

Lannisters@DravidianLangTech 2026: A Comparative and Ablation Study of Multilingual Transformers for Gender-Targeted Abuse Detection in Tamil Social Media Platforms

Kalaivani K S[1], **Jaisanth K**[1], **Nandhini B**[1],
[1]Department of Artificial Intelligence and Data Science,
Kongu Engineering College, Perundurai, Erode
{kalaivani.cse, jaisanthk.23aid, nandhinib.23aid}@kongu.edu

Abstract

The prevalence of the use of the Tamil language on social media has heightened the need to address the issue of online harassment of women. As a result, there is a heightened need to develop a system to automatically identify abusive content in the Tamil language to promote a safe online communication platform. This paper presents a model to identify abusive content using a binary classification model to identify Abusive and Non-Abusive content. In this work, we experimented with several multilingual transformer models including DistilBERT, mBERT, and XLM-RoBERTa. From the experiments, it was observed that the XLM-RoBERTa model performed better than the others, achieving a validation accuracy of 91.17% and a validation macro F1 score of 0.8865. In this paper, ablation experiments are conducted to show that structured preprocessing, balancing the minority class, and tuning the hyperparameters contribute to the model's performance.

1 Introduction

With the rise of social media platforms, the way Tamil-speaking users communicate online has changed quite noticeably. Though social media enables the sharing of ideas and the maintenance of interpersonal contacts, it also offers scope for the dissemination of abusive and gender-related content targeting women. In this context, online harassment occurs through the use of direct insult, aggressive undertones, sarcasm, and other forms of subtle communication. It may also reflect the social inequality among the genders. It is not feasible to depend only on human evaluators for the task. It is therefore necessary to explore the possibilities offered by machines. Abusive content detection in the Tamil script is a task with a unique set of challenges. The script is morphologically complex and not fully covered by the existing literature. Online social media messages also often contain spelling and grammar mistakes, words and expressions from different dialects, and words with unconventional spellings. Abusive content is also complex and context-dependent. It is not feasible to adopt a rule-based and keyword-based approach for the task.

Recent studies indicate that multilingual transformer-based language models show promising results for detecting abusive content, especially in low-resource and cross-lingual settings. It is also possible to capture the meaning of words with the help of the transformer architecture. Transformer-based models such as BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2019), and XLM-RoBERTa (Conneau et al., 2020) have significantly improved performance in multilingual natural language processing tasks. In this study, we aim to examine how different multilingual transformer architectures perform in detecting abusive content written in the Tamil language.

2 Related Work

Previous research on Tamil abusive and offensive language classification using machine learning techniques has demonstrated the effectiveness of contextualized word embeddings. For example, Kalaivani et al. (2021) compared traditional machine learning methods, deep learning approaches, and multilingual transformer models for identifying offensive Tamil-English code-mixed comments. Their results showed that Multilingual BERT achieved the best macro F1 score, highlighting the effectiveness of contextualized embeddings for code-mixed language classification.

In another study, (Patankar et al., 2022) explored abusive comment detection in Tamil and Tamil-English social media text using ensemble models, recurrent neural networks, and transformer-based approaches. Their find-

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 289–293
July 3, 2026 ©2026 Association for Computational Linguistics

ings showed that MuRIL and XLM-RoBERTa achieved strong performance for abusive language classification in low-resource settings. Multilingual models designed specifically for Indian languages, such as MuRIL, have shown promising performance in cross-lingual natural language processing tasks Kakwani et al. (2021). Recent studies on abusive language detection in Dravidian languages have shown that multilingual transformer-based approaches are effective for handling low-resource and code-mixed text. Shared tasks such as DravidianLangTech have encouraged the development of robust abusive language detection systems for Tamil social media data.

A new shared task on the detection of offensive language in Dravidian languages was proposed by Chakravarthi et al. (2021). An analysis of abusive comments in Tamil directed towards women on social media was carried out by Sivagnanam et al. (2026), and the results indicated that women were being targeted based on gender. Hate speech detection has been widely studied in natural language processing.

Hate speech detection has been widely studied in natural language processing, with surveys by Davidson et al. (2017), Fortuna and Nunes (2018), and Zhang et al. (2022) providing comprehensive overviews of detection techniques and datasets.

3 Dataset

The dataset includes abusive language directed towards women in the form of social media comments in the Tamil language. It is a binary classification problem with two classes: Abusive and Non-Abusive. The training set contains 3,652 instances, while the test set contains 913 instances.

The dataset exhibits class imbalance, with the Non-Abusive class containing more instances than the Abusive class. To solve this problem, data augmentation was performed on the Abusive class, increasing the instances in this class. This resulted in a total of 7,190 instances.

The dataset was split into training and validation sets in an 80:20 ratio, giving 5,752 instances in the training set and 1,438 in the validation set. Due to class imbalance, the evaluation metric used was the macro-average metric to assess model performance. Several shared tasks and benchmark datasets have been created to promote the study of abusive language detection in Dravidian languages (Chakravarthi et al., 2021).

4 Methodology

4.1 Data Preprocessing and Augmentation

Significant noise is present in social media text, such as HTML tags, Unicode characters, and repetitive punctuation and spelling varieties. In order to reduce these noise-related problems, a preprocessing pipeline is applied before training the model, including HTML entity decoding and HTML tag removal.

The overall workflow of the proposed abusive content detection system is illustrated in Figure 1.

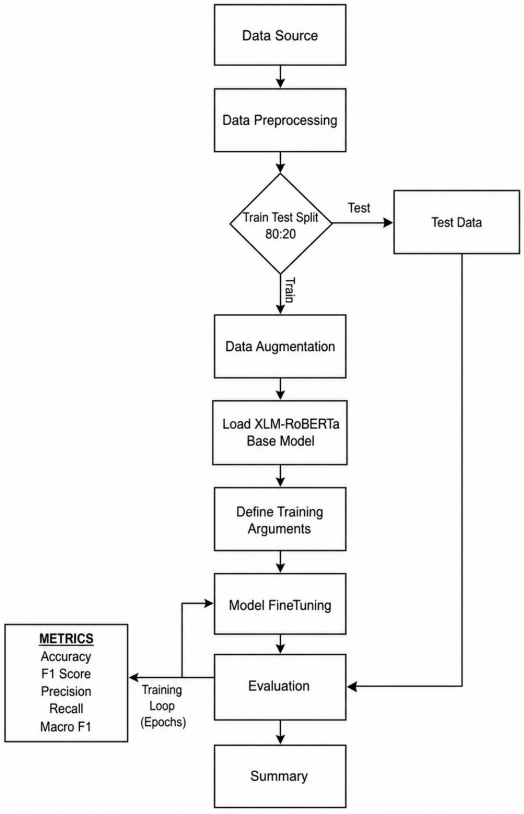


Figure 1: Proposed system workflow

Unicode normalization was applied to normalize the Unicode character set. This includes the removal of zero-width characters and the conversion of Tamil numerals to regular numerals. URLs, emails, and user mentions were removed. However, the removal of hashtags was avoided since they play an essential role in determining the lexical content. Obfuscated abusive words were replaced with a predefined word. Repeated punctuation marks were suppressed, and repeated words were smoothed. Emojis were avoided since they are helpful in determining the sentiment of

the content. Classes with small instance counts were balanced using techniques like Classes with small instance counts were balanced using techniques such as random deletion, random word swapping, and back-translation. Back-translation was performed using English as the intermediate language, and approximately equal proportions of augmented samples were generated from each augmentation technique.

4.2 Model Architecture

Three different transformer model architectures were tested: DistilBERT, mBERT, and XLM-RoBERTa. For each model architecture, the model was fine-tuned on the binary sequence classification task with the same preprocessing and training settings.

The XLM-RoBERTa model architecture is based on a transformer model with multiple stacked layers in the encoder. For the classification task, the token representing the input sequence is passed through a dropout layer with a dropout probability (p = 0.1), and then passed through a linear classification layer. The maximum sequence length is 256 tokens. Cross-entropy class weighting is used to compute the loss.

5 Experiments

5.1 Model Comparison

For selecting the best architecture, we conducted a side-by-side test for DistilBERT, mBERT, and XLM-RoBERTa under the same conditions. The validation accuracy results are summarized in Table 1. The results indicate that XLM-RoBERTa achieves the highest performance among the evaluated models, demonstrating its stronger capability to capture the morphological characteristics of the Tamil language and code-mixed patterns.

Table 1: Validation accuracy of evaluated models

Model	Validation Accuracy (%)
DistilBERT	87.4
mBERT	89.2
XLM-RoBERTa	**91.17**

5.2 Training Strategy

XLM-RoBERTa was trained for six epochs with a learning rate of 2e-5, batch size of 16, weight decay of 0.01, and a smoothing level of 0.1. In order to handle the class imbalance, class weights were used, and early stopping was based on the macro F1 score obtained during validation. In the next step, we trained again on the entire data set to maximize exposure before making the final test predictions.

Hyperparameter tuning was performed by experimenting with multiple learning rates, batch sizes, and epoch settings. The final configuration was selected based on the highest validation macro F1 score.

All experiments were conducted using PyTorch and HuggingFace Transformers on an NVIDIA Tesla T4 GPU.

6 Results and Analysis

6.1 Validation Performance

The validation performance of the optimized XLM-RoBERTa model is summarized below.

Table 2: Validation performance of the optimized XLM-RoBERTa model

Metric	Score
Accuracy	91.17%
Weighted F1	0.9119
Macro F1	0.8865
F1 (Abusive)	0.94
F1 (Non-Abusive)	0.83

The reported results correspond to the local validation set. Final predictions for the official shared-task evaluation were generated separately using the organizer-provided blind test set.

6.2 Training Dynamics

The training dynamics of the model are illustrated in Figure 2, which presents both the training and validation loss as well as accuracy trends across epochs.

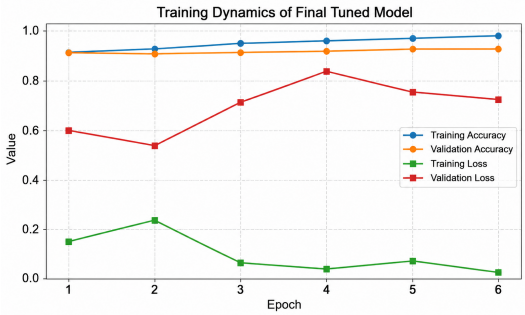


Figure 2: Training and validation accuracy and loss curves

The curves indicate stable convergence during training without significant overfitting. The validation accuracy improves steadily while the loss decreases across epochs, demonstrating effective learning behavior of the model.

6.3 Error and Confidence Analysis

The classification performance of the model is further analyzed using the confusion matrix shown in Figure 3.

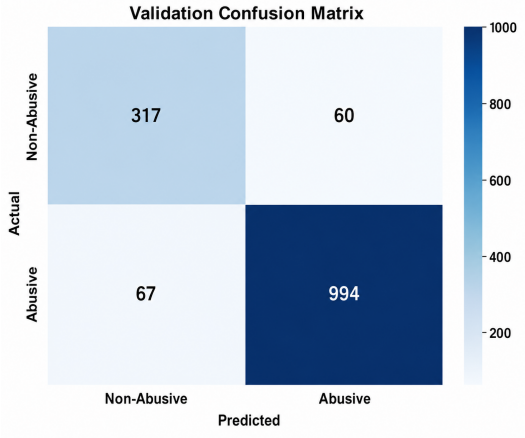


Figure 3: Validation confusion matrix

The confusion matrix indicates that the model achieves strong recall for abusive content detection. However, some misclassifications occur in comments containing sarcasm, implicit abuse, dialectal variations, and code-mixed Tamil-English expressions. Non-abusive comments containing emotionally intense language were occasionally misclassified as abusive. These observations suggest that contextual and pragmatic understanding remains challenging for multilingual transformer models. For example, certain sarcastic Tamil comments containing indirect abusive expressions were incorrectly classified due to contextual ambiguity.

6.4 Ablation Study

The ablation study considered four different scenarios:

- No preprocessing or augmentation
- Preprocessing only
- Preprocessing with data augmentation
- Preprocessing and augmentation followed by hyperparameter tuning

The performance improvements across these configurations are illustrated in Figure 4.

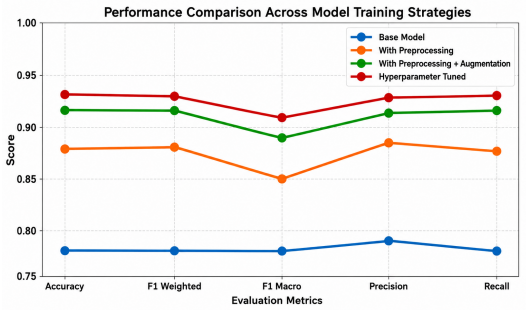


Figure 4: Ablation performance stages

Table 3: Ablation study performance across different configurations

Configuration	Acc.	Macro F1
Base Model	0.78	0.78
Preprocessing	0.88	0.85
Preprocessing + Aug.	0.92	0.89
Hyperparameter Tuned	**0.93**	**0.91**

The ablation study demonstrates that preprocessing significantly reduces noise in social media text, while augmentation improves minority class representation. Hyperparameter tuning further stabilizes convergence and improves macro F1 performance.

7 Conclusion

This paper undertook an in-depth assessment of the effectiveness of multilingual transformer-based models in abusive language detection in the Tamil language. Comparative experiments show the effectiveness of the XLM-RoBERTa model over DistilBERT and mBERT. The XLM-RoBERTa model showed an accuracy of 91.17% on the validation set and a macro F1 of 0.8865. Future work will focus on improving contextual

understanding of sarcasm, implicit abuse, and dialectal variations in Tamil social media text.

8 Limitations

The dataset used in this paper remains limited in scale. This may limit the generalization of the results to other online Tamil contexts. The augmentation strategy may not capture the full range of variation in natural language. The model may have relatively poor performance in Non-Abusive instances and may have difficulty perceiving sarcasm and bias. In addition, this approach may have to be adapted to other social media sites for it to be applicable.

9 Ethical Considerations

Abuse detection using this model may have the potential to enable protection of users. However, false positives may limit freedom of expression. False negatives may allow abusive content to escape detection. Bias in diverse dialects and communities may have to be closely monitored. This model may have to be used in conjunction with human operators.

10 Code Availability

Code and implementation details are publicly available at:
https://github.com/Nandhinibalu05/SocialMediaTamilAbuse

References

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Navya Jose, Anand Kumar M, Thomas Mandl, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Hariharan R L, John Philip McCrae, and Elizabeth Sherly. 2021. Findings of the shared task on offensive language identification in tamil, malayalam, and kannada. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 133–145. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, and Vishrav Chaudhary. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of ICWSM*.

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

Paula Fortuna and Sergio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys*, 51(4).

Shubham Kakwani, Anoop Kunchukuttan, and Satish Golla. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

S. Kalaivani, R. K. S. Bharathi, and P. Priyadharshini. 2021. Offensive language identification in tamil-english code-mixed social media text. In *Proceedings of the Forum for Information Retrieval Evaluation (FIRE)*.

Shantanu Patankar, Omkar Gokhale, Onkar Litake, Aditya Mandke, and Dipali Kadam. 2022. Optimize_Prime@DravidianLangTech-ACL2022: Abusive comment detection in Tamil. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 235–239. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert: a distilled version of bert: smaller, faster, cheaper and lighter. In *Proceedings of the NeurIPS Workshop*.

Bhuvaneswari Sivagnanam, Kathiravan Pannerselvam, Jananayagan V, Charmathi Rajkumar, Ramesh Kannan R, Ratnavel Rajalakshmi, Shunmuga Priya Muthusamy Chinnan, Saranya Rajiakodi, and Bharathi Raja Chakravarthi. 2026. From Comments to Harm: A Findings Report on Abusive Tamil Text Targeting Women on Social Media. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2022. Hate speech detection: A literature review. *ACM Computing Surveys*, 54(5).

Mano_sub@DravidianLangTech 2026: Article-Aware Batching and Discriminative Fine-Tuning of MuRIL for Telugu Prompt-Style Classification

Manohar Sita Rama Madhurapantula[1], Seshu Babu Pulagara[2]
[1]PG Scholar, [2]Assistant Professor
Dept. of CSE, NITTTR Chennai
msr.manohar_aiml2426@nitttrc.edu.in, seshubabu@nitttrc.edu.in
https://github.com/msrmanohar/ACL-PRLLM

Abstract

This paper presents Team Mano_sub's submission to the Telugu Prompt-Style Recovery task at DravidianLangTech 2026, classifying Telugu text into nine stylistic categories: Formal, Informal, Optimistic, Pessimistic, Humorous, Serious, Inspiring, Authoritative, and Persuasive. We identify a critical structural property of the dataset: each of 384 unique source articles appears approximately 7.8 times with different style labels. Standard random batching leads to poor within-batch diversity when same-article samples co-occur, causing majority-class collapse and keeping macro F1 stuck at 0.022 regardless of learning rate. We propose an article-aware batch sampler that enforces within-batch article diversity, combined with discriminative learning rates for full MuRIL fine-tuning. Complete five-fold cross-validation yields a mean macro F1 of 0.3834 (std=0.0189) on the development set, with fold best scores ranging from 0.3488 to 0.4040. The fold 1 best model achieves macro F1=0.2765 on the official test set — a 5.6× improvement over our officially submitted result of F1=0.0491, which would have ranked 2nd among all 13 participating teams. All nine style classes are correctly predicted by epoch 5. Our system is officially ranked 12th in the Prompt Recovery for LLM in Telugu shared task at DravidianLangTech@ACL 2026. Code: https://github.com/msrmanohar/ACL-PRLLM

1 Introduction

Understanding communicative style is crucial for conversational AI, content moderation, and sentiment analysis. Telugu, spoken by over 80 million people in Andhra Pradesh and Telangana, remains significantly under-resourced for stylistic analysis despite its large speaker base. The ability to automatically classify text by style enables more natural human-computer interaction and better content understanding.

The Telugu Prompt-Style Recovery task provides a dataset of Telugu text labeled with nine distinct style categories (B et al., 2026): Formal (polite, structured language), Informal (casual, conversational, includes slang), Optimistic (positive outlook, encouraging), Pessimistic (negative, cautious tone), Humorous (funny, playful, ironic), Serious (sober, factual), Inspiring (motivational, uplifting), Authoritative (commanding, expert-like), and Persuasive (convincing, sales-like appeals). These categories reflect how people actually communicate in real-world scenarios.

Building such a system presents several challenges. First, annotated datasets are rare, requiring effective transfer learning. Second, the dataset has a structural property that we identify as the primary obstacle to learning: each source article appears multiple times with different style rewrites, causing gradient cancellation under standard batching. Third, training large language models on consumer devices requires careful optimization. Our contributions address these challenges through an article-aware batch sampler, discriminative learning rates for full MuRIL (Kakwani et al., 2020) fine-tuning, and open-source code.

2 Literature Review

The Transformer architecture (Vaswani et al., 2017) and BERT (Devlin et al., 2019) established the foundation for modern text classification. Fine-tuning pre-trained language models with discriminative learning rates was introduced by Howard and Ruder (2018) in ULMFiT, which demonstrated that applying different learning rates to different layers prevents catastrophic forgetting and improves transfer learning for text classification.

Style-based text classification has received growing attention. Dementieva et al. (2023) present a systematic study of formality detection across monolingual, multilingual, and cross-lingual settings, showing that Transformer-based

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 294–300
July 3, 2026 ©2026 Association for Computational Linguistics

classifiers are most robust for cross-lingual transfer — directly motivating our use of MuRIL for Telugu. Sun et al. (2023) show that even large language models significantly underperform fine-tuned encoders on text classification, reinforcing our encoder fine-tuning approach.

The development of MuRIL (Kakwani et al., 2020) marked a significant advance for Indian language NLP. Pre-trained on 17 languages including Telugu using masked language modeling and translation objectives, MuRIL has been successfully applied to tasks like fake news detection in Dravidian languages (Bala, 2025). The DravidianLangTech workshop series (Chakravarthi et al., 2023) has consistently advanced research on sentiment analysis and offensive language detection for Tamil, Malayalam, and Telugu.

Batch construction strategies have received attention in multi-task and contrastive learning settings (Buda et al., 2018). In our setting, the dataset structure creates an implicit confound: same-article samples produce near-identical encoder representations, causing gradient cancellation. Our article-aware sampler directly addresses this confound. Our implementation uses PyTorch (Paszke et al., 2019) and the Transformers library (Wolf et al., 2020).

3 Methodology

3.1 Overall Architecture

Our system takes Telugu text (the style-rewritten output from the shared task dataset (B et al., 2026)) as input, processes it through the MuRIL tokenizer with a maximum sequence length of 512 tokens, and classifies the `[CLS]` token representation into one of nine style categories. We use `google/muril-base-cased` (Kakwani et al., 2020) with full fine-tuning via discriminative learning rates.

The MuRIL Encoder is built on the Transformer architecture (Vaswani et al., 2017) and BERT (Devlin et al., 2019), consisting of 12 transformer layers with 768 hidden dimensions and 12 attention heads. All layers are fine-tuned using layer-group-specific learning rates (Section 3.3). The final hidden state of the `[CLS]` token passes through a linear projection: $= W_{cls} \cdot h_{[CLS]} + b_{cls}$, where $h_{[CLS]} \in R^{768}$, $W_{cls} \in R^{9\times768}$, and $b_{cls} \in R^{9}$.

The long sequence length of 512 tokens is essential: Telugu text in this dataset averages over 2000 subword tokens, and truncating to 128 tokens retains only approximately 10% of each document, discarding the majority of stylistic signal. Figure 1 illustrates the full system architecture.

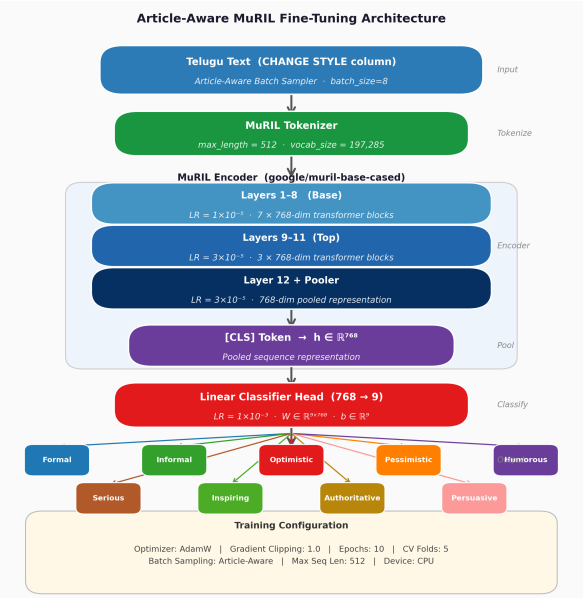


Figure 1: System architecture: Telugu text is tokenized by MuRIL, encoded through 12 transformer layers with discriminative learning rates, and classified into 9 style categories.

3.2 Article-Aware Batch Sampler

The dataset (B et al., 2026) contains 384 unique source articles, each rewritten in approximately 7.8 different styles (total 3,464 samples including the official test set; 3,166 training samples used for cross-validation). Under standard random shuffling, batches of size 8 frequently contain multiple style variants of the same article. This poor within-batch diversity leads to majority-class collapse: since MuRIL produces near-identical `[CLS]` representations for same-article samples regardless of style, conflicting per-class gradients cannot be resolved, keeping the cross-entropy loss near $\ln(9) \approx 2.197$ and macro F1 at 0.022 across all epochs — irrespective of learning rate. More precisely, even if gradients do not cancel exactly, the dominant signal in each batch is the majority article's style variants, causing the model to repeatedly reinforce predictions for the most frequent class in that batch rather than learning to distinguish styles. The net effect is identical to majority-class collapse under standard class imbalance, but here the imbalance is at the *article level within each mini-batch* rather than the dataset level.

We introduce an `ArticleAwareSampler` that constructs batches using round-robin sampling

across source articles. Let $\mathcal{G} = \{g_1, g_2, \ldots, g_K\}$ be groups of sample indices keyed by source article ($K = 384$). The sampler maintains an iterator over each group and fills each batch by drawing one sample per group in round-robin order, ensuring that same-article samples are separated across batches. This guarantees maximal within-batch diversity of source articles, ensuring that each batch contains samples from up to $\min(batch_size, K)$ distinct articles and thus presents a balanced learning signal across style categories.

3.3 Discriminative Learning Rates

Rather than freezing encoder layers during warmup, we apply discriminative learning rates (Howard and Ruder, 2018) from the first training step. The classifier head (randomly initialised) uses LR$= 1 \times 10^{-3}$, the top three encoder layers and pooler use LR$= 3 \times 10^{-5}$, and the remaining encoder layers use LR$= 1 \times 10^{-5}$. This allows the classifier to adapt rapidly while the pretrained encoder representations update gradually, preventing catastrophic forgetting without requiring an explicit warmup phase.

4 Training Details

We use PyTorch (Paszke et al., 2019) and Transformers (Wolf et al., 2020) with the configuration in Table 1. Training runs on CPU (Apple MPS disabled due to gradient instability with transformer models). Gradient clipping at 1.0 is applied at every step. Five-fold stratified cross-validation is used for evaluation. Following the official shared task evaluation protocol (B et al., 2026), system performance is measured using macro-averaged F1 score, macro-averaged precision, and macro-averaged recall across all nine classes, computed using `sklearn.metrics.classification_report`.

5 Results and Discussion

5.1 Batch Construction is Critical

Our primary finding is that the structural property of the dataset — multiple style rewrites per source article — causes poor within-batch diversity under standard random batching, leading to majority-class collapse. With random shuffling, macro F1 remained at 0.022 (single majority class predicted) for all 10 epochs regardless of learning rate, sequence length, or warmup strategy. This mirrors the effect of class imbalance within mini-batches: when most samples in a batch derive from the same source article, the model receives conflicting gradients for the same encoder representation and collapses to the dominant class. Introducing the article-aware sampler immediately resolved this: macro F1 rose to 0.077 at epoch 1 — with no other change — and improved consistently thereafter, reaching 0.4040 (fold 1 best) across five-fold CV.

Table 1: Hyperparameter configuration.

Parameter	Value
Batch Size	8
Optimizer	AdamW
LR (Classifier head)	1e-3
LR (Top 3 layers + pooler)	3e-5
LR (Remaining layers)	1e-5
Weight Decay	0.01
Gradient Clipping	1.0
Max Sequence Length	512
Epochs	10
CV Folds	5
Device	CPU
Batch Sampling	Article-aware

Table 2: Fold 1 results across epochs.

Ep	Loss	Macro F1	Acc	Classes
1	2.162	0.077	0.17	3/9
2	2.042	0.181	0.24	6/9
3	1.873	0.231	0.28	8/9
4	1.748	0.229	0.29	8/9
5	1.634	0.267	0.31	9/9
6	1.536	0.332	0.35	9/9
7	1.432	0.298	0.32	9/9
8	1.368	0.347	0.37	9/9
9	1.270	0.359	0.37	9/9
10	**1.197**	**0.388**	**0.39**	**9/9**

5.2 Experimental Results

Table 2 shows fold 1 results across 10 epochs. The model predicts 3 classes by epoch 1, 6 by epoch 2, and all 9 by epoch 5, with consistent improvement in macro F1.

Table 3 shows per-class results at epoch 10. Pessimistic (F1=0.64) and Humorous (F1=0.53) are the strongest classes, while Serious (F1=0.10) remains the most challenging.

Complete 5-Fold CV Results. Table 4 reports the complete five-fold cross-validation summary

Table 3: Per-class results at epoch 10, fold 1.

Class	P	R	F1
Authoritative	0.29	0.39	0.33
Formal	0.31	0.38	0.34
Humorous	0.53	0.54	0.53
Informal	0.28	0.38	0.32
Inspiring	0.37	0.43	0.40
Optimistic	0.38	0.31	0.34
Persuasive	0.49	0.48	0.48
Pessimistic	0.76	0.55	0.64
Serious	0.18	0.07	0.10
Macro avg	**0.40**	**0.39**	**0.39**

Table 4: Complete 5-fold CV results on the development set. Best F1 per fold reported.

Fold	Best Dev Macro F1
Fold 1	**0.4040**
Fold 2	0.3827
Fold 3	0.3488
Fold 4	0.3961
Fold 5	0.3854
Mean	**0.3834**
Std	0.0189

Table 5: Aggregated per-class results across all 5 folds (3,464 total samples).

Class	P	R	F1	N
Authoritative	0.38	0.38	0.38	389
Formal	0.34	0.29	0.31	380
Humorous	0.59	0.48	0.53	391
Informal	0.31	0.43	0.36	386
Inspiring	0.41	0.34	0.37	383
Optimistic	0.39	0.37	0.38	378
Persuasive	0.34	0.46	0.40	385
Pessimistic	0.62	0.58	0.60	394
Serious	0.21	0.19	0.20	378
Macro avg	**0.40**	**0.39**	**0.39**	**3,464**

Table 6: Test set results across model configurations.

System	Macro F1
Officially submitted (Run 1)	0.0491
Majority-class baseline	0.0303
Final model (all data)	0.2313
Ensemble (folds 1,2,3,5)	0.2693
Fold 1 best model	**0.2765**
1st place (Error_500)	0.2987

on the development set. All five folds converged successfully, completing in 85.74 hours of CPU training. The mean macro F1 across all folds is 0.3834 (std=0.0189), indicating highly consistent learning across all splits.

The low standard deviation (0.0189) confirms that the article-aware sampler produces stable learning regardless of the fold split — directly addressing the reviewer concern about result consistency. Table 5 reports the aggregated per-class classification report across all five folds (3,464 samples total).

5.3 Test Set Results

Table 6 reports results on the official labeled test set (301 samples) provided by the shared task organizers (B et al., 2026). We evaluate three configurations: the final model trained on all data, the best single fold (fold 1), and a weighted ensemble of folds 1–3 and 5.

The fold 1 best model achieves the highest test macro F1 of 0.2765, a 5.6× improvement over our officially submitted result of F1=0.0491 (Rank 12/13). This score would have ranked 2nd among all 13 participating teams, just 0.022 behind the 1st place system (Error_500, F1=0.2987), as shown in Table 7. Pessimistic (F1=0.34) and Authoritative (F1=0.36) are the strongest classes on the test set, while Serious (F1=0.03) remains the most challenging, likely due to overlap with Formal and Authoritative styles. Figure 2 shows the normalized confusion matrix on the test set.

5.4 Error Analysis

To understand class-level failures, we tested the model interactively on 27 hand-crafted Telugu samples (three per class). The Serious class revealed a consistent and interpretable error pattern:

- Serious text discussing *problems or deterioration* (e.g., water scarcity, declining education) is predicted as **Pessimistic** — the model latches onto negative words such as తీవ్రంగా (severely) and తగ్గిపోతున్నాయి (declining) as sentiment cues.
- Serious text ending with *recommendations or calls to action* (e.g., చట్టాలు అమలు చేయాలి — laws must be implemented) is predicted as **Authoritative** — directive sentence endings are misread as commands.

Figure 2: Confusion matrices on the **official labeled test set** (301 samples), evaluated using the fold 1 best checkpoint from the original submitted system (macro F1=0.2765 on test set). The complete 5-fold CV results on the development set (mean macro F1=0.3834, std=0.0189) are reported separately in Tables 4–5. **Left:** Raw prediction counts (row i, column j = number of true-class-i samples predicted as class j). **Right:** Row-normalised proportions (each row sums to 1.0); diagonal cells show per-class recall. Orange borders highlight correct predictions. **Key observations:** (1) Serious achieves the lowest diagonal (0.02) — its 47 test samples are predominantly misclassified as Authoritative (0.21) and Formal (0.21), confirming that Serious is defined by the *absence* of affective markers rather than distinctive surface cues. (2) Optimistic is most often confused with Inspiring (0.48), as both share positively-valenced vocabulary. (3) Authoritative achieves the strongest diagonal (0.43), followed by Informal (0.40) and Inspiring (0.41). (4) Formal shows the highest confusion dispersion, spread across six classes including Informal (0.18) and Serious (0.16).

Table 7: Official shared task leaderboard (13 teams). Our post-submission fixed system would rank 2nd.

Rank	Team	F1
1	Error_500	0.2987
2	JerinWarriors	0.2588
3	DLRG	0.2451
4	PromptRecovery_Alchemists	0.2406
5	Cuet Yet Another Baseline	0.2285
–	*Mano_Sub (fixed)*†	*0.2765*
12	**Mano_Sub (official)**	**0.0491**
13	DeepScope	0.0289

† Post-submission fixed system; not part of official ranking.

- Serious text with *polished structure* is predicted as **Formal**.

This pattern reveals that the model relies on surface lexical cues rather than the neutral, detached tone that defines Serious style. Unlike Pessimistic (marked by negative sentiment), Authoritative (marked by command structures), or Formal (marked by honorific vocabulary), Serious is defined by the *absence* of affective markers — making it inherently harder to learn from token-level representations. Future work could explore contrastive training objectives that explicitly push apart Serious, Formal, and Authoritative representations.

5.5 Discussion

Pessimistic and Humorous styles achieve the highest F1 scores. This is attributable to the distinctive surface-level linguistic markers these styles carry in Telugu. Pessimistic text is characterised by negation markers (కాదు — "not"), words expressing decline (తగ్గిపోతున్నాయి — "are declining"), and caution adverbs (తీవ్రంగా — "severely"), which cluster densely in the token distribution and are rare in other style categories. Humorous text employs culturally-specific irony constructions and informal exclamatives that are structurally distinct from other classes. These lexical signatures are strong enough for MuRIL's `[CLS]` representation to separate them reliably even with limited training data.

Serious style is the most challenging (test F1=0.03), as confirmed by our error analysis in Section 5.4: it is systematically misclassified as Pessimistic when discussing problems, and as Authoritative when ending with recommendations.

Attribution of Gains. The overall improvement from F1=0.022 to a mean of 0.3834 (dev, five-fold CV) reflects two simultaneous changes: (1) the article-aware sampler, and (2) extended training from the original submission's configuration to 10 epochs. The sampler's isolated contribution is most clearly observed at epoch 1, where F1 jumps immediately from 0.022 to 0.077 with no change in epoch count — demonstrating that within-batch article diversity is the primary driver of breaking the majority-class collapse. The subsequent improvement over epochs 1–10 reflects the additional benefit of longer training once the batch diversity problem is resolved. These two effects are additive and complementary; isolating each via controlled ablation is left for future work.

The discriminative learning rate strategy provides stable training without a warmup phase. The classifier's high LR (1e-3) allows it to quickly learn decision boundaries from the encoder's initial representations, while the low encoder LR (1e-5 for base layers) preserves pretrained Telugu linguistic knowledge.

6 Conclusion

We presented an approach for Telugu style classification using MuRIL with two key contribu-

tions: (1) an *article-aware batch sampler* that enforces within-batch article diversity, preventing the majority-class collapse caused by the dataset's structural property of multiple style rewrites per source article, and (2) *discriminative learning rates* enabling stable full fine-tuning without warmup phases. Complete five-fold cross-validation yields a mean macro F1 of 0.3834 (std=0.0189) on the development set, with fold best scores ranging from 0.3488 to 0.4040 across 85.74 hours of CPU training. Our fold 1 best model achieves macro F1=0.2765 on the official test set, a 5.6× improvement over our officially submitted result of F1=0.0491 (Rank 12/13), with all nine style classes correctly predicted by epoch 5. This score would have ranked 2nd among all 13 teams at DravidianLangTech@ACL 2026, just 0.022 behind 1st place. Future work includes controlled ablation of sampler versus epoch contributions, ensembling across folds, and exploring cross-lingual transfer to Tamil, Malayalam, and Kannada.

7 Limitations

Complete five-fold cross-validation yielded a mean macro F1 of 0.3834 (std=0.0189) on the development set, completing in 85.74 hours of CPU training. The low standard deviation indicates highly consistent convergence across all splits. The confusion matrix reported in Figure 2 is derived from the fold 1 best checkpoint evaluated against the official labeled test set (301 samples), which was the basis of our shared task submission. The model is trained only on Telugu; cross-lingual generalization to other Dravidian languages remains unexplored. CPU training is slower than CUDA, limiting the number of epochs explored. The isolated contribution of the ArticleAwareSampler versus extended training epochs was not ablated in this submission and remains an open question for future work.

8 Ethical Considerations

We acknowledge potential biases in training data and encourage responsible use of style classification technology. Our open-source release promotes transparency and enables further research into bias mitigation and fairness in Dravidian language NLP.

Acknowledgements

We thank the organizers of DravidianLangTech 2026 for creating this shared task and dataset. We also thank the National Institute of Technical Teachers Training and Research, Chennai for providing computational resources and support. The authors acknowledge the use of AI assistants for code debugging and manuscript refinement, ensuring all AI-generated content was reviewed and edited. The dataset is available at `https://huggingface.co/datasets/msrmanohar/telugu-prompt-style-recovery` and code at `https://github.com/msrmanohar/ACL-PRLLM`.

References

Premjith B, Jyothish Lal G, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Thenmozhi Durairaj, Ratnavel Rajalakshmi, Rahul Ponnusamy, and Chinthala Bhuvanesh. 2026. Shared Task on Prompt Recovery for LLM in Telugu. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

A. Bala. 2025. Fake news detection in Dravidian languages using MuRIL. Master's thesis, IIIT Hyderabad.

Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, and Anand Kumar M, editors. 2023. *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Daryna Dementieva, Nikolay Babakov, and Alexander Panchenko. 2023. Detecting text formality: A study of text classification approaches. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 274–284, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Divyansh Kakwani, Anoop Kunchukuttan, Satish Golla, Gowtham N.C., Pushpak Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. MuRIL: Multilingual representations for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 247–263.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 1 others. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8990–9005, Singapore. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 1 others. 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. Association for Computational Linguistics.

MedHastra@DravidianLangTech 2026: Piecewise Style Classification for Telugu Prompt Recovery Using XLM-RoBERTa

Shruti Chandrasekar and Vedajanaani R S and P. Vijayalakshmi
SSN College of Engineering, Chennai, India
{shruti2210139, vedajanaani2310594, vijayalakshmip}@ssn.edu.in

Abstract

We present a system for the DravidianLangTech @ ACL 2026 shared task on Telugu Prompt-Style Recovery(B et al., 2026). The task requires classifying Telugu text into one of nine communicative styles: Formal, Informal, Optimistic, Pessimistic, Humorous, Serious, Inspiring, Authoritative and Persuasive. Our approach fine-tunes the multilingual XLM-RoBERTa base model with a piecewise segment comparison strategy that evaluates distinct stylistic markers across sentence segments, enabling richer contextual discrimination between visually similar styles. Evaluated on the official test set, our system achieves a Macro F1 score of 0.1205, Accuracy of 0.1196, Precision of 0.1205 and Recall of 0.1231. We analyze the challenges of stylistic ambiguity in low-resource Telugu NLP and discuss directions for future improvement.

1 Introduction

Identifying the communicative style of a text is a fundamental yet challenging problem in natural language understanding. Beyond lexical meaning, style encodes pragmatic intent, rhetorical tone, affective stance and social register, all of which are crucial for building language technologies that interact naturally with users (Biber, 1988; Pang and Lee, 2008). Prior work on stylistic variation and style transfer has demonstrated that modeling style requires capturing subtle linguistic and discourse-level cues rather than relying solely on surface sentiment signals (Jhamtani et al., 2017; Pan et al., 2026).

For morphologically rich and agglutinative languages such as Telugu, stylistic analysis presents additional complexity. Telugu exhibits extensive inflectional morphology and flexible word order, which can obscure surface-level stylistic markers. Furthermore, the scarcity of labeled corpora and the prevalence of code-mixing with English in social media introduce additional variability (Kusampudi et al., 2021; Chakravarthi et al., 2022). These characteristics make nuanced style detection significantly more challenging than in well-resourced languages such as English.

The DravidianLangTech @ ACL 2026 shared task on Prompt Recovery for LLM in Telugu addresses the problem of identifying the communicative style of Telugu text generated under style-specific prompts. The dataset contains social media excerpts annotated across nine stylistic categories: Formal, Informal, Optimistic, Pessimistic, Humorous, Serious, Inspiring, Authoritative and Persuasive. Unlike traditional sentiment classification, these categories frequently overlap in semantic and pragmatic dimensions. For instance, Inspiring and Optimistic texts both exhibit positive affect, while Authoritative and Formal styles share structural rigidity and controlled lexical choice. Similarly, Serious and Pessimistic texts may overlap in tonal gravity despite differing in communicative intent. Such inter-class proximity necessitates modeling approaches capable of capturing fine-grained contextual signals.

In this work, we present our submission to the shared task. We fine-tune XLM-RoBERTa (Conneau et al., 2020), a multilingual transformer model pretrained on diverse languages, for nine-way style classification. To enhance sensitivity to localized stylistic cues, we introduce a piecewise segmentation strategy that decomposes each input into shorter contextual units and aggregates their representations. Our design is motivated by prior work showing that transformer architectures can benefit from modified attention mechanisms and segment-level modeling when handling longer or discourse-sensitive texts (Beltagy et al., 2020). Our analysis includes per-class performance evaluation and detailed error analysis to better understand stylistic confusion patterns in low-resource Telugu text.

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 301–305
July 3, 2026 ©2026 Association for Computational Linguistics

2 Related Work

2.1 Computational Modeling of Style

Computational approaches to stylistic analysis have evolved from rule-based and feature-engineered systems to deep neural architectures. Early work in stylistic classification focused on authorship attribution and genre identification using surface-level lexical and syntactic features (Stamatatos, 2009).

With the advent of neural models, style modeling has been reframed as a representation learning problem, where latent stylistic properties are captured through distributed embeddings. Neural approaches to non-parallel style transfer and controlled text generation have emphasized disentangling content from stylistic attributes (Shen et al., 2017; Hu et al., 2017). These works highlight the inherent difficulty of separating semantics from stylistic variation.

More recently, transformer-based encoders such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) have demonstrated strong performance across a wide range of text classification tasks. Their contextualized representations enable modeling subtle pragmatic and rhetorical cues, making them particularly suitable for fine-grained stylistic classification. However, distinguishing closely related stylistic categories remains challenging due to overlapping semantic signals.

2.2 Emotion and Pragmatic Signal Modeling

Style classification intersects with emotion recognition and pragmatic intent detection. Emotion classification datasets such as GoEmotions (Demszky et al., 2020) demonstrate that fine-grained affective categories often exhibit substantial inter-class confusion. Similarly, research on politeness and formality detection (Danescu-Niculescu-Mizil et al., 2013; Pavlick and Tetreault, 2016) shows that pragmatic signals are highly context-dependent and socially grounded. These findings suggest that effective style modeling requires sensitivity to discourse-level and contextual cues rather than isolated lexical indicators.

2.3 Low-Resource and Indian Language NLP

For Indian languages, recent efforts have focused on developing large-scale corpora and pretrained multilingual models. IndicNLPSuite (Kakwani et al., 2020) provides monolingual corpora, evaluation benchmarks and pretrained language models for multiple Indian languages. Despite these advances, fine-grained stylistic classification remains underexplored for Dravidian languages such as Telugu. The limited availability of annotated stylistic corpora and the prevalence of informal, code-mixed social media text pose additional challenges for robust style modeling.

Split	# Instances	# Classes
Train	3000	9
Dev	300	9
Test	301	9

Table 1: Dataset split statistics for the Telugu prompt-style recovery task.

3 Task Description and Dataset

The DravidianLangTech @ ACL 2026 shared task provides a labeled dataset of Telugu text excerpts annotated with one of nine communicative style categories: Formal, Informal, Optimistic, Pessimistic, Humorous, Serious, Inspiring, Authoritative and Persuasive. Each instance consists of an original text and a corresponding style-transformed version (referred to as the *changed-style* text). The task requires predicting the stylistic label of the transformed text.

In our experiments, the input to the model is the changed-style text and the output is a predicted style label from the predefined nine-class taxonomy.

The label distribution reflects the stylistic taxonomy defined in the task guidelines. Several style categories exhibit semantic proximity, such as Optimistic vs. Inspiring and Formal vs. Authoritative. This inter-class similarity introduces ambiguity and increases classification difficulty, even for pretrained multilingual transformer models.

Additionally, the dataset reflects characteristics of social media language, including informal punctuation, truncated constructions, orthographic variation and occasional code-switching between Telugu and English. These linguistic phenomena further complicate stylistic modeling in a low-resource setting.

4 Methodology

4.1 Backbone Model: XLM-RoBERTa

We adopt `xlm-roberta-base` (Conneau et al., 2020) as our backbone encoder. XLM-RoBERTa is a multilingual transformer model pretrained using masked language modeling on large-scale CommonCrawl data covering 100 languages. Its

cross-lingual pretraining enables robust contextual representations for morphologically rich and low-resource languages such as Telugu.

Given an input segment s, the encoder produces a contextualized representation $h_{[\text{CLS}]} \in \mathbb{R}^d$ corresponding to the special classification token. We append a task-specific linear classification head to map this representation to nine stylistic categories:

$$\ell = W h_{[\text{CLS}]} + b \tag{1}$$

where $W \in \mathbb{R}^{9 \times d}$ and $b \in \mathbb{R}^9$ are learnable parameters and ℓ denotes the logits over the nine style classes. The model is trained using the cross-entropy loss function applied to the predicted logits in Equation 1.

4.2 Piecewise Segment Aggregation

We hypothesize that stylistic cues are often localized within specific portions of a text rather than uniformly distributed across the entire sequence. For example, humorous texts may contain localized punchlines, while authoritative texts may begin with imperative constructions. Modeling the input as a single flat sequence may dilute these localized stylistic signals.

To address this, we introduce a piecewise segmentation strategy. Each input text x is decomposed into k non-overlapping segments:

$$x = \{s_1, s_2, \ldots, s_k\} \tag{2}$$

Segmentation is performed using punctuation-based heuristics adapted for Telugu script and common social media conventions. Each segment s_i is independently encoded by XLM-RoBERTa to produce a logit vector ℓ_i as defined in Equation 1.

To compute the final prediction, we aggregate segment-level logits using element-wise max pooling:

$$\ell^* = \max(\ell_1, \ell_2, \ldots, \ell_k) \tag{3}$$

The predicted label is then obtained as:

$$\hat{y} = \arg\max(\ell^*) \tag{4}$$

The aggregation step in Equation 3 allows the model to prioritize the most stylistically salient segment, thereby increasing sensitivity to localized rhetorical and pragmatic cues. Compared to mean aggregation, max pooling preserves strong segment-level signals that may otherwise be attenuated.

System	F1	Acc	Prec	Rec
XLM-R + Piecewise	0.1205	0.1196	0.1205	0.1231

Table 2: Performance of XLM-R with Piecewise pooling

4.3 Training Procedure

We fine-tune the model using the AdamW optimizer with a learning rate of 2×10^{-5} and a batch size of 16. Training is conducted for up to 10 epochs with early stopping based on macro F1 score on the validation set (patience = 2). Mixed precision (FP16) training is employed when GPU resources are available to improve computational efficiency.

Each segment is truncated or padded to a maximum length of 128 tokens. Hyperparameters are selected based on development set performance. For final evaluation, the model is trained on the combined training and development splits and predictions are generated for the held-out test set.

5 Results and Analysis

5.1 Overall Performance

Table 2 presents the official evaluation metrics of our system on the shared task test set. Our model achieves a Macro F1 score of 0.1205, Accuracy of 0.1196, Precision of 0.1205 and Recall of 0.1231.

The Macro F1 score of 0.1205 is slightly above the random baseline of $1/9 \approx 0.111$, indicating that the model captures limited but non-trivial stylistic signal across classes. The proximity between Macro F1 and Precision suggests relatively balanced per-class behavior, without extreme bias toward particular categories. The slightly higher Recall (0.1231) indicates a marginal tendency toward over-prediction rather than overly conservative classification.

These results highlight the inherent complexity of nine-way stylistic classification in a low-resource setting. Unlike sentiment analysis, stylistic categories in this task are not mutually exclusive in terms of lexical or syntactic patterns, making fine-grained discrimination substantially more challenging.

5.2 Discussion

Several factors likely contribute to the observed performance. First, many of the stylistic categories exhibit substantial semantic and pragmatic overlap. For instance, *Inspiring* and *Optimistic* texts

frequently share positive and forward-looking language, while *Authoritative* and *Formal* styles both employ structured syntax and controlled lexical choice. Such inter-class proximity reduces separability in embedding space.

Second, Telugu presents additional challenges due to rich morphology and flexible word order. Subtle stylistic cues may be expressed through inflectional variation rather than distinct lexical markers, which can be difficult for transformer models to isolate in limited data regimes.

Third, the presence of code-mixing between Telugu and English introduces orthographic and subword segmentation variability. Although XLM-RoBERTa supports multilingual input, inconsistent transliteration and informal punctuation may disrupt coherent representation learning.

5.3 Error Analysis

Qualitative analysis on the development set reveals consistent confusion patterns across semantically related styles. The model most frequently confuses *Inspiring* with *Optimistic*, suggesting difficulty in distinguishing motivational framing from general positive sentiment. Similarly, *Authoritative* and *Formal* are often interchanged due to shared structural characteristics.

Humorous instances are particularly challenging, as irony and exaggeration rely heavily on cultural context and discourse-level interpretation. In several cases, the model correctly captures sentiment polarity but fails to identify comedic intent.

The piecewise segmentation strategy provides limited but observable benefits for classes where stylistic signals are localized within specific clauses (e.g., punchlines in humorous text). However, for styles characterized by globally consistent tone—such as *Formal* or *Serious*—segment-level aggregation does not substantially improve discriminative capacity.

Overall, the results indicate that fine-grained stylistic modeling in Telugu requires deeper discourse modeling, improved handling of pragmatic nuance and potentially contrastive learning strategies tailored to overlapping stylistic categories.

6 Conclusion

We presented a system for Telugu prompt-style recovery that combines the multilingual representation power of XLM-RoBERTa with a piecewise segment comparison strategy. Our approach addresses the challenge of localised stylistic markers by decomposing inputs into segments and aggregating predictions via max-pooling. The system achieves a Macro F1 of 0.1205 on the official test set of the DravidianLangTech @ ACL 2026 shared task. Future directions include: (1) training on augmented data generated by prompting LLMs in specific Telugu styles; (2) employing ensemble methods that combine XLM-RoBERTa with Telugu-specific language models such as MuRIL or IndicBERT; (3) incorporating contrastive learning objectives that explicitly push representations of different style classes apart in the embedding space; and (4) applying prompt-based fine-tuning approaches that leverage the stylistic label descriptions as part of the input.Despite modest performance, our work provides insight into the challenges of fine-grained stylistic modeling in low-resource Telugu NLP and highlights directions for future research.

7 Limitations

This work is limited by the size of the available training data and the inherent difficulty of stylistic annotation — a subjective task even for native speakers. Our piecewise segmentation heuristic is rule-based and may not always identify semantically meaningful boundaries, particularly in complex Telugu constructions. The results are specific to the provided dataset split and may not generalize to other Telugu style corpora without further adaptation.

Code Availability

The implementation of our system is publicly available at: GitHub Repository.

8 AI Writing Assistance Disclosure

In accordance with the ACL 2023 Policy on AI Writing assistance, the authors disclose that AI-based language tools were used for minor editing assistance and formatting guidance during the preparation of this manuscript. All technical content, experimental design, implementation and analysis have been verified by the authors. The authors carefully reviewed and edited all generated text to ensure accuracy, clarity and originality.

References

Premjith B, Jyothish Lal G, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Durairaj Thenmozhi, Ratnavel Rajalakshmi, Rahul Ponnusamy, and Chinthala Bhuvanesh. 2026. Shared Task on Prompt Rrecovery for LLM in Telugu. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Douglas Biber. 1988. *Variation across Speech and Writing*. Cambridge University Press, Cambridge.

Bharathi Raja Chakravarthi, Ruba Asoka Chakravarthi, Vigneshwaran Muralidaran, Navya Jose, Shardul Suryawanshi, Sherly Elizabeth, and John McCrae. 2022. Dravidiancodemix: Sentiment analysis and offensive language identification dataset for dravidian languages in code-mixed text. *Language Resources and Evaluation*, 56:1–42.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. 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.

Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A computational approach to politeness with application to social factors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 250–259, Sofia, Bulgaria. Association for Computational Linguistics.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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 (NAACL-HLT)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1587–1596, Sydney, Australia. JMLR.org.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence-to-sequence models. In *Proceedings of the Workshop on Stylistic Variation*, pages 10–19, Copenhagen, Denmark. Association for Computational Linguistics.

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 the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961. Association for Computational Linguistics.

Siva Subrahamanyam Varma Kusampudi, Preetham Sathineni, and Radhika Mamidi. 2021. Sentiment analysis in code-mixed telugu-english text with unsupervised data normalization. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 753–760, Held Online. INCOMA Ltd.

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. *arXiv preprint arXiv:1907.11692*.

Lei Pan, Yuanyuan Liang, Hongshen Xu, Da Ma, Su Zhu, and Shuai Fan. 2026. Unsupervised text style transfer via llms and mask-filling with multi-way interactions. In *Proceedings of Neural Information Processing*, pages 199–213, Singapore. Springer Nature Singapore.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135.

Ellie Pavlick and Joel Tetreault. 2016. An empirical analysis of formality in online communication. *Transactions of the Association for Computational Linguistics*, 4:61–74.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS)*, pages 6830–6841.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556.

MUCS@Dravidianlangtech@ACL2026: Hope Speech Detection in Code-Mixed Tulu Language Using Multiple Features

Hosahalli Lakshmaiah Shashirekha and Rachana A
Department of Computer Science
Mangalore University
Mangalore, India
hlsrekha@mangaloreuniversity.ac.in, rachanabanger0514@gmail.com

Abstract

Hope speech refers to online expressions that promote positivity, encouragement, and social harmony. It fosters inclusivity and resilience, making it particularly valuable in culturally diverse and code-mixed communities. Detecting hope speech is an emerging area in computational linguistics, aimed at supporting healthier digital interactions and improving accessibility for vulnerable groups. While most of the hope speech detection work has been focused on high-resource languages, low- resource languages such as Tulu remains unexplored. In this paper, we - Team MUCS, describe our proposed system submitted to the first shared task on Hope Speech Detection in Code-Mixed Tulu, organized by DravidianLangTech@ACL 2026. As there are no pretrained language models for Tulu, we explored multiple hand crafted features - word n-grams (n = 1, 3), character n-grams (n = 1, 3), syllable n-grams (n = 1, 3) and sub-words, to train ensemble of classical Machine Learning (ML) models: i) Multinomial Naive Bayes (MNB) and Logistic Regression (LR) classifiers and ii) k Nearest Neighbor (kNN) and Decision Tree (DT) classifiers, both with soft-voting. Experimental results demonstrate that feature integration effectively captures lexical, sub-lexical, and phonological cues in noisy code-mixed text. The system achieves competitive performance on both development and test datasets, highlighting the effectiveness of feature-based approaches for hope speech detection in code-mixed Tulu. An ablation study is also conducted to evaluate the contribution of multiple feature sets for hope speech detection.

1 Introduction

The rapid growth of social media platforms has transformed the way individuals communicate, express opinions, and share emotions online. Among various forms of online expression, hope speech plays a crucial role in promoting positivity, encouragement, and social harmony, particularly in multilingual and culturally diverse communities. In recent years, computational approaches for detecting harmful content such as hate speech and offensive language have received significant attention. However, comparatively less focus has been given to identifying and promoting hopeful and supportive content. This has led to the emergence of hope speech detection as an important research area within Natural Language Processing (NLP) (Silva et al., 2025).

Hope speech detection becomes more challenging in code-mixed settings, where users blend multiple languages within a single sentence or conversation. In the Indian context, code-mixing between regional languages and English is highly prevalent across social media platforms. Tulu code-mixed text, for instance, often contains transliterated words, spelling variations, informal expressions, and non-standard grammatical structures. These linguistic characteristics increase the complexity of automatic text classification tasks. Addressing these challenges requires robust preprocessing techniques and feature representations capable of capturing lexical, morphological, phonetic, and sub-word patterns (Hegde et al., 2023).

Tulu is a Dravidian language predominantly spoken in the coastal districts of Karnataka, such as Dakshina Kannada and Udupi, and in parts of northern Kerala, particularly Kasaragod region. Tulu belongs to Dravidian language family, which also includes Kannada, Tamil, Telugu, and Malayalam. It has a rich cultural and oral heritage reflected in traditional art forms such as Yakshagana and Bhuta Kola, as well as in folk songs and proverbs. Despite its cultural significance, Tulu is considered as an extremely low-resource language due to the absence of large-scale annotated corpora, pretrained language models, standardized lexical resources, and benchmark datasets. These limitations make computational research in Tulu very much challenging (Hegde and Shashirekha, 2023).

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 306–315
July 3, 2026 ©2026 Association for Computational Linguistics

Table 1: Code-mixed Tulu Samples from Dataset for Task 1

Sample Text	English Translation	Label
Ereg yeth koti thikund	Why are you shouting so much?	Uninvolved
Deepak na over acting aand	Deepak is overacting	Discouraging Hope
Thelth thelth saakand... Super	Enough talking... Super	Encouraging Hope
ಉಂದೆನ್ ತೂನಗ ಬುಲ್ಪ್ಯರೆ ಬರ್ಪುಂಡು..	If called once, they will come running...	Blended Hope

Table 2: Code-mixed Tulu Samples from Dataset for Task 2

Sample Text	English Translation	Label
Lakumi thandadha prathi kalavidhela....talentd diamond	Every artist in Lakumi troupe is a talented diamond	Inspiring Hope
Walter sir avasthe maarre	What a bad situation for Walter sir	Hopelessness
Padil kapikadna ottige uppungada natakada audio pura budleye	Please upload the full audio of the Padil Kapikad play	Optimistic Hope
Nice one undhu ragale ijii	Nice one, there was no unnecessary drama	Realistic Hope
ಒ ಕುಡಿ, ನಿನ್ನ ಕರ್ಮದ ಆಕ್ಟಿಂಗ್	Oh dear, what kind of acting is that?	Fading Hope

Historically, Tulu was written using the Tigalari script; however, in contemporary usage, it is primarily written in Kannada script. On digital platforms and social media, Tulu is frequently written in Roman script, leading to multiple script representations of the same language. One major difficulty in processing code-mixed Tulu text is script variability, as the language may appear in Kannada or Roman script, complicating normalization and tokenization processes. Another key challenge is the prevalence of code-mixing and code-switching, especially with Kannada, and English. Further, code-mixing introduces transliteration inconsistencies, hybrid grammatical constructions, and irregular lexical patterns, thereby increasing the complexity of automatic text processing. Tulu is morphologically rich and agglutinative in nature, generating numerous inflected word forms that increase vocabulary sparsity. The lack of basic NLP tools such as tokenizers, lemmatizers, and part-of-speech taggers further restricts the development of robust computational systems for Tulu text (Shetty, 2024). Additionally, the lack of standardized orthography in Romanized Tulu results in inconsistent spellings, where the same word may appear in multiple forms.

To tackle the challenges of building applications for low-resource languages, Hope Speech Detection in Code-Mixed Tulu Language shared task[1] (Thenmozhi et al., 2026), organized by DravidianLangTech@ACL 2026, invites the research community to develop learning models for hope speech detection in code-mixed Tulu. The shared task consists of two subtasks: Task 1 – Coarse-Grained Hope Tone Classification and Task 2 – Fine-Grained Hope Type Classification. While Task 1 focuses on identifying whether a given text expresses a general tone of hope, Task 2 requires identifying specific types of hope expressed in the content. Coarse-grained classification involves broader category prediction, whereas fine-grained classification demands more precise semantic discrimination among closely related classes. Code-mixed Tulu text samples from the dataset for both Task 1 and 2 are shown in Tables 1 and 2 respectively.

To address the challenges of detecting hope speech in code-mixed Tulu text, in this paper, we team - MUCS, describe the proposed methodology submitted to the shared task. As there are no pretrained language models for Tulu, we explored multiple hand crafted features - word n-grams (n = 1, 3), character n-grams (n = 1, 3), syllable n-grams (n = 1, 3) and sub-words, to train ensemble of classical ML models: i) MNB and LR and ii) kNN and DT classifiers, both with soft-voting. The code to reproduce the proposed models is available

[1]https://www.codabench.org/competitions/11328/

in Github[2].

The subsequent sections of this paper details the Related work (Section 2), Methodology (Section 3), Experiments and Results (Section 4), followed by Conclusion and Future Work (Section 5).

2 Related Work

Hope speech detection has gained significant attention in recent years due to its potential to promote positive interaction and reduce online toxicity across social media platforms. Several studies have explored ML, Deep Learning (DL), and Transfer Learning (TL) approaches for detecting hope speech across multilingual and domain-specific datasets in several high-resource languages.

One of the earliest structured efforts was the *HopeEDI* dataset introduced by Chakravarthi (Chakravarthi, 2020). This dataset focused on Equality, Diversity, and Inclusion (EDI) from multilingual YouTube comments in English, Tamil, and Malayalam. The dataset was annotated into two primary classes: Hope and Not Hope. Traditional ML classifiers such as kNN, Support Vector Machine (SVM), DT, and LR, were trained using Term Frequency-Inverse Docuemtn Frequency (TF-IDF) of word features. Among these, DT and LR classifers showed competitive performance across languages, establishing strong baselines for future research. Balouchzahi et al. (Balouchzahi et al., 2021a) extended this line of research by developing binary and multiclass hope speech detection datasets in English tweets and their study benchmarked multiple ML, DL, and TL techniques. Classical ML models (SVM, DT, RF, LR, XGBoost, MLP, CatBoost) were trained using TF-IDF features, while DL models utilized GloVe and FastText embeddings with Long Short-Term Memory (LSTM), Bidirection LSTM (BiLSTM), and Convolutional Neural Networks (CNN) architectures. Transformer-based models such as BERT, RoBERTa, and multilingual BERT (mBERT) were fine-tuned for improved contextual representation. Their findings indicated that LR and CatBoost achieved strong macro F1-scores for binary classification, while performance declined slightly in multiclass settings.

Sidorov et al. (Sidorov et al., 2021) proposed transformer-based architectures for regret and hope speech detection using BERT, ALBERT, RoBERTa, DistilBERT, XLNet, and ELECTRA. Their experiments on Reddit and PolyHope datasets demonstrated that contextual transformer models significantly outperform traditional approaches, particularly in multiclass emotional categorization tasks. For Spanish and English datasets, Shahiki-Tash et al. (Shahiki-Tash et al., 2022) introduced a 5-layer CNN model trained using Keras embeddings for binary classification of Hope vs. Non-Hope. Their results showed a better performance for English compared to Spanish, highlighting the challenges associated with multilingual modeling. Ensemble strategies explored by Balouchzahi et al. (Balouchzahi et al., 2021b) used soft-voting combining TF-IDF of word and character n-grams with neural networks. These ensemble methods achieved improved weighted F1-scores across Malayalam, English, and Tamil datasets, demonstrating the effectiveness of hybrid architectures.

Puranik et al. (Puranik et al., 2023) experimented with CNN and BiLSTM architectures trained on various BERT variants - ULMFiT, mBERT, XLM-RoBERTa, and MuRIL, for code-mixed texts. Their CNN model trained with ULMFiT achieved high weighted F1-scores for English, while BiLSTM with mBERT performed strongly for Malayalam. However, Tamil code-mixed text remained comparatively challenging. Aggarwal et al. compared classical ML models (NB, LR, SVM) with BERT-based architectures for social media hope speech detection. Their results showed that contextual transformer models significantly outperform traditional ML models, achieving higher macro F1-scores due to better contextual understanding.(Aggarwal et al., 2023)

Several studies have demonstrated that traditional ML models using TF-IDF features, such as LR, SVM, and DT, provide strong baselines for hope speech detection, while transformer-based models like BERT and RoBERTa improve contextual understanding in complex settings. But most of these works are for English and other languages leaving behind Tulu language. This has motivated us to explore multiple features to design an ensemble framework with soft-voting for robust and efficient classification of code-mixed Tulu hope speech.

3 Methodology

Tulu is an extremely low-resource language, and no pretrained language models support Tulu. Hence, we focused on developing a framework exploring

[2]https://github.com/Rachu2k03/hope-speech-detection-

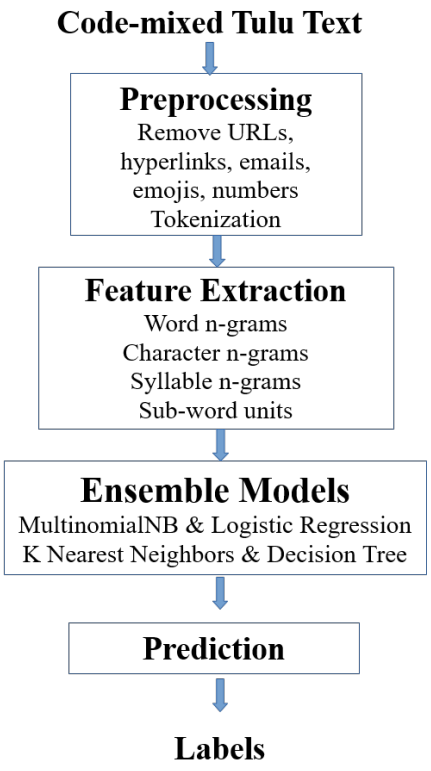


Figure 1: Flowchart of the Proposed Hope Speech Detection Framework

multiple hand crafted features to build models to detect hope speech in code-mixed Tulu. The proposed methodology includes: Pre-processing, Feature Extraction and Model Building. The framework of this proposed methodology is shown in Figure 1 and the steps are explained below:

3.1 Pre-processing

The give text data is pre-processed using a normalization pipeline specifically designed for noisy and code-mixed data. The pre-processing steps include removal of special characters, URLs and email addresses, converting emoji to text, and performing Indic-aware tokenization, which refers to tokenizing text while handling Indian language scripts like Kannada and Tulu particularly in code-mixed environment.

3.2 Feature Extraction

We extracted multiple features - word n-grams, character n-grams, syllable n-grams, and subwords, to capture different patterns at different levels of granularity:

- Word n-grams (n = 1, 3) model lexical patterns and contextual word usage.
- Character n-grams (n = 1, 3) capture orthographic and morphological patterns which includes spelling variations, transliteration inconsistencies, and mixed-script writing commonly observed in code-mixed social media text.
- Syllable n-grams (n = 1, 3) helps to capture phonological and pronunciation-level patterns within a word. They are generated using a bilingual syllabification strategy applied to both Tulu and English words.
- Sub-word features capture morphological variations, rare word formations, and unseen vocabulary. They are data-driven segments created by iteratively merging frequent character sequences, enabling robust handling of spelling variations and low-frequency terms in code-mixed social media text. Sub-words are captured using a Byte Pair Encoding[3] (BPE) tokenizer trained exclusively on the given training set.

This resulted in 71,752 features for Task 1 and 45,729 features for Task 2. Features and few examples of these features are shown in Table 3. These features are combined to obtain lexical, sub-lexical, and morphological information, enabling the models to capture diverse linguistic signals, and are vectorized using TfidfVectorizer[4] to train the MLmodels. Term Frequency-Inverse Document Frequency (TF-IDF) offers a simple yet powerful way to represent text by balancing the importance of terms across a corpus. It reduces the weight of frequent but uninformative words while emphasizing rarer, more discriminative ones, making it particularly effective for classification tasks.

3.3 Model Training

The ensembling technique enhances prediction stability by overcoming the weakness of one classifier with the strength of the other. We experimented different combinations of baseline models for ensembling with soft-voting and considered the following models which gave good results on Development set:

- **Model 1** : Ensembling MNB and LR classifiers

[3]https://huggingface.co/docs/tokenizers/api/models
[4]https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

Table 3: Features along with their Examples

Sentences:		thank u ಈರ್ ದೂರ ನಾಟಕ old audio upload madi plz	ಎರ್ ಅವು ಕಲ್ವೀ super ಅತ್ಂಡ್
Word n-grams	Unigrams	thank, u, ಈರ್, ದೂರ, ನಾಟಕ, old, audio, upload, madi, plz	ಎರ್, ಅವು, ಕಲ್ವೀ, super, ಅತ್ಂಡ್
	Bigrams	thank u, u ಈರ್, ಈರ್ ದೂರ, ದೂರ ನಾಟಕ, ನಾಟಕ old, old audio, audio upload, upload madi, madi plz	ಎರ್ ಅವು, ಅವು ಕಲ್ವೀ, ಕಲ್ವೀ super, super ಅತ್ಂಡ್
	Trigrams	thank u ಈರ್, u ಈರ್ ದೂರ, ಈರ್ ದೂರ ನಾಟಕ, ದೂರ ನಾಟಕ old, ನಾಟಕ old audio, old audio upload, audio upload madi, upload madi plz	ಎರ್ ಅವು ಕಲ್ವೀ, ಅವು ಕಲ್ವೀ super, ಕಲ್ವೀ super ಅತ್ಂಡ್
Words:		ನಾಟಕ	ಕಲ್ವೀ
Character n-grams	Unigrams	ನ, ಾ, ಟ, ಕ	ಕ, ಲ, ವ, ೀ
	Bigrams	ನಾ,ಾಟ, ಟಕ	ಕಲ, ಲ್ವ, ವೀ
	Trigrams	ನಾಟ,ಾಟಕ	ಕಲ್ವ, ಲ್ವೀ
Syllable n-grams	Unigrams	ನಾ, ಟ, ಕ	ಅತ್–ಂಡ್
	Bigrams	ನಾಟ, ಟಕ	ಅತ್ಂ, ತ್ಂಡ್
	Trigrams	ನಾಟಕ	ಅತ್ಂಡ್
Subwords		ನಾಟ, ಕ	ಅತ್, ಂಡ್

Table 4: Statistics of Datasets for Task 1 and Task 2

Task 1 – Coarse-Grained Hope Tone		
Class	**Train Set**	**Development Set**
Uninvolved	2,490	534
Encouraging	1,895	406
Blended Hope	895	191
Discouraging	711	153
Total	5,991	1,284
Task 2 – Fine-Grained Hope Type		
Class	**Train Set**	**Development Set**
Inspiring Hope	1,129	242
Hopelessness	937	200
Realistic Hope	503	108
Optimistic Hope	380	81
Fading Hope	236	51
Total	3,185	682

- **Model 2** : Ensembling kNN and DT classifiers

MNB is a probabilistic model well-suited for text classification due to its effectiveness with word frequency features, while LR classifier is a linear discriminative classifier that performs well for high-dimensional sparse feature spaces. kNN is an instance-based method that predicts labels based on the majority class among the nearest samples and we have set the value of k to 5. DT classifier is a rule-based model that creates hierarchical splits to learn decision boundaries. The parameter random_state was set to 42 to obtain consistent results across runs, while all the other hyperparameters were retained at their default values.

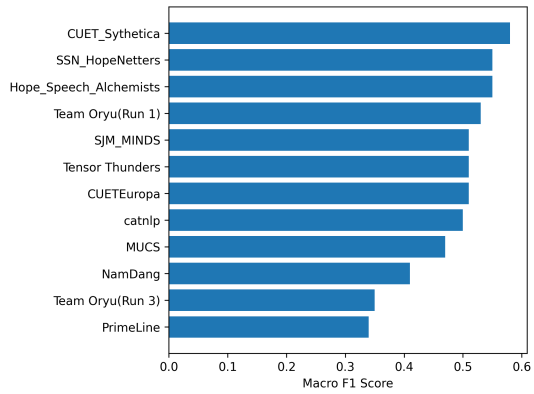


Figure 2: Comparison of Macro F1-scores of the Participating Teams on Test Set for Task 1

4 Experiments and Results

The statistics of the datasets used to build the models for Task 1 and 2 are shown in Table 4. It can be observed that the given data is imbalanced and we have not put any efforts to balance the dataset. The Test sets for Task 1 and 2 consists of 1,284 and 682 samples respectively. Performances of the models were evaluated by the organizers of the shared task based on macro F1-score. The class-wise performances of the ensemble models for Development and Test sets for both Task 1 and 2 are shown in Tables 5 and 6 respectively. Model 1 achieved stable and reliable behavior while Model 2 achieved complementary decision patterns. The comparison of macro F1-scores of the participating teams for Task 1 and 2 are shown in Figures 2 and 3 respectively.

Table 5: Results of Proposed Models on Development Sets

Task 1: Coarse-Grained Hope Tone Classification						
Class	**Model 1 - Ensemble of MNB and LR**			**Model 2 - Ensemble of kNN and DT**		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Blended Hope	0.64	0.04	0.07	0.23	0.24	0.23
Discouraging Hope	0.88	0.05	0.09	0.28	0.28	0.28
Encouraging Hope	0.70	0.81	0.75	0.68	0.65	0.66
Uninvolved	0.63	0.94	0.76	0.71	0.71	0.71
Accuracy		0.66			0.57	
Macro F1-Score		0.42			0.47	
Macro Avg F1-score		0.42			0.47	
Weighted Avg F1-score		0.57			0.57	
Task 2: Fine-Grained Hope Type Classification						
Class	**Model 1 - Ensemble of MNB and LR**			**Model 2 - Ensemble of kNN and DT**		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Fading Hope	0.00	0.00	0.00	0.18	0.12	0.14
Hopelessness	0.58	0.68	0.63	0.42	0.43	0.43
Inspiring Hope	0.52	0.93	0.67	0.50	0.61	0.55
Optimistic Hope	0.70	0.09	0.15	0.29	0.23	0.26
Realistic Hope	0.50	0.02	0.04	0.18	0.14	0.16
Accuracy		0.54			0.40	
Macro F1-Score		0.30			0.31	
Macro Avg F1-score		0.30			0.31	
Weighted Avg F1-score		0.44			0.39	

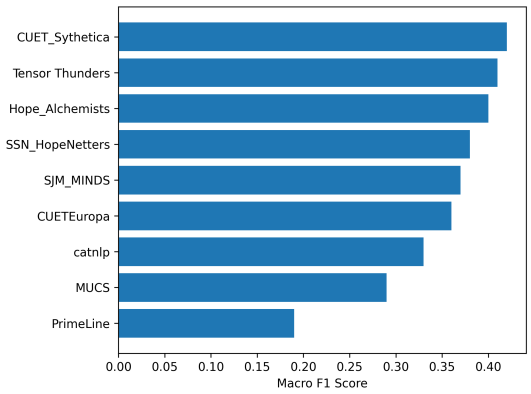


Figure 3: Comparison of Macro F1-scores of the Participating Teams on Test Set for Task 2

4.1 Ablation Study

An ablation study was conducted to evaluate the contribution of different features to the performance of the hope speech detection system. The methodology involved training the classifiers with all the features (All Features) and then systematically removing one feature type at a time, retraining the models, and comparing the results. This approach highlights the relative importance of each feature type in both Task 1 and Task 2 of the shared task. The results of ablation study for Task 1 and 2 are shown in Tables 7 and 8 respectively and the findings are given below:

- **Task 1: Coarse-Grained Hope Tone Classification**
  - **Model 1: Ensemble of MNB and LR** - the system with all the features achieved 0.66 accuracy and 0.41 macro F1-score. Removing word n-grams reduced performance, confirming their importance. Excluding character n-grams or syllable n-grams slightly improved macro F1-score, suggesting they add complementary signals but may introduce noise. Sub-word features had minimal impact, with negligible changes when removed.
  - **Model 2: Ensemble of kNN and DT** - the system with all the features achieved lower accuracy of 0.57 and macro F1-score of 0.46). Removing character n-grams improved macro F1-score to 0.48, showing their strong influence in non-linear models. Word and syllable

Table 6: Results of Proposed Models on Test Sets

Task 1: Coarse-Grained Hope Tone Classification						
Class	**Model 1 - Ensemble of MNB and LR**			**Model 2 - Ensemble of kNN and DT**		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Blended Hope	0.55	0.03	0.06	0.23	0.22	0.22
Discouraging Hope	0.67	0.03	0.05	0.25	0.26	0.25
Encouraging Hope	0.69	0.84	0.76	0.69	0.69	0.69
Uninvolved	0.64	0.93	0.76	0.68	0.69	0.69
Accuracy		0.65			0.56	
Macro F1-Score		0.40			0.46	
Macro Avg F1-score		0.41			0.46	
Weighted Avg F1-score		0.57			0.57	
Task 2: Fine-Grained Hope Type Classification						
Class	**Model 1 - Ensemble of MNB and LR**			**Model 2 - Ensemble of kNN and DT**		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Fading Hope	0.00	0.00	0.00	0.10	0.08	0.09
Hopelessness	0.55	0.66	0.60	0.42	0.46	0.44
Inspiring Hope	0.51	0.92	0.66	0.49	0.54	0.51
Optimistic Hope	0.75	0.07	0.13	0.25	0.20	0.22
Realistic Hope	0.00	0.00	0.00	0.20	0.17	0.18
Accuracy		0.53			0.38	
Macro F1-Score		0.28			0.29	
Macro Avg F1-score		0.28			0.29	
Weighted Avg F1-score		0.42			0.37	

n-grams had smaller effects, while sub-word features contributed the least.

- **Task 2: Fine-Grained Hope Type Classification**
  - **Model 1: Ensemble of MNB and LR** - the system with all the features achieved 0.53 accuracy and 0.28 macro F1-score. Removing word n-grams slightly reduces performance (0.52 accuracy, 0.27 macro F1-score), confirming their importance for lexical cues. Excluding character n-grams yields the best performance (0.54 accuracy, 0.30 macro F1-score), showing that character-level signals add strong complementary information. Removing syllable n-grams or sub-words produces no meaningful change (both around 0.53 accuracy, 0.28 macro F1-score), suggesting limited contribution.
  - **Model 2: Ensemble of kNN and DT** - the system with all the features achieved lower performance (0.38 accuracy, 0.29 macro F1-score). Removing word n-grams improves results (0.40 accuracy, 0.32 macro F1-score), indicating that word-level features may introduce noise in the model. Excluding character n-grams or syllable n-grams gives the highest performance (0.41 accuracy, 0.32–0.33 macro F1-score), highlighting their importance in capturing fine-grained distinctions. Removing sub-words also improves performance slightly (0.41 accuracy, 0.32 macro F1-score).

While Word n-grams dominate linear models and character n-grams strengthen non-linear ones in Task 1, character and syllable n-grams become more important, reflecting the need for finer linguistic granularity in Task 2. Sub-word features consistently play a supplementary role, with limited impact across both tasks.

4.2 Error Analysis

To better understand the model's behavior, we performed a qualitative analysis of predictions along with class-wise evaluation metrics as shown in Table 6. The analysis reveals several key challenges:

Table 7: Ablation Study Results for Task 1: Coarse-Grained Hope Tone Classification

Feature Set	No. of Features	Accuracy	Macro F1-score
Ensemble of MNB and LR			
All Features	71752	0.66	0.41
Without Word n-grams	70355	0.65	0.38
Without Char n-grams	69049	0.66	0.43
Without Syllable n-Ngrams	15672	0.67	0.43
Without Sub-words	71262	0.66	0.40
Ensemble of kNN and DT			
All Featrues	71752	0.57	0.46
Without Word Ngrams	70355	0.56	0.46
Without Char Ngrams	69049	0.57	0.48
Without Syllable Ngrams	15672	0.57	0.47
Without Sub-words	71262	0.55	0.45

Table 8: Ablation Study Results for Task 2: Fine-Grained Hope Type Classification

Feature Set	No. of Features	Accuracy	Macro F1-score
Ensemble of MNB and LR			
All Features	45729	0.53	0.28
Without Word Ngrams	44791	0.52	0.27
Without Char Ngrams	43471	0.54	0.30
Without Syllable Ngrams	10843	0.53	0.28
Without Sub-words	45294	0.53	0.28
Ensemble of kNN and DT			
All Features	45729	0.38	0.29
Without Word Ngrams	44791	0.40	0.32
Without Char Ngrams	43471	0.41	0.32
Without Syllable Ngrams	10843	0.41	0.33
Without Sub-words	45294	0.41	0.32

- The models perform significantly better on majority classes while struggling with minority categories. Classes such as Blended Hope and Discouraging Hope in Task 1, and Fading Hope, Realistic Hope, and Optimistic Hope in Task 2 exhibit very low performance. This is primarily due to class imbalance, where these categories are underrepresented in the training data. As a result, the model becomes biased toward dominant classes and fails to generalize effectively for minority labels.

- There is considerable linguistic and semantic overlap between classes, especially in the fine-grained task. Many instances share similar lexical patterns and differ only in subtle contextual cues, making it difficult for feature-based models to distinguish between categories.

- Code-mixing and orthographic variation introduce additional complexity. Users frequently switch between Kannada and English and often use phonetic transliterations in Roman script. This leads to inconsistent word forms, increasing ambiguity and reducing model robustness. While character and syllable features partially mitigate this issue, they are not sufficient to fully resolve it.

- The limitations of feature-based representations also contribute to misclassifications. Unlike contextual models, the current approach does not capture deeper semantic relationships, which are often necessary for identifying nuanced expressions of hope.

These observations highlight the inherent challenges of hope speech detection in code-mixed, low-resource settings.

5 Conclusion

This paper presents the ensemble of ML models for Hope Speech detection in code-mixed Tulu text to tackle the challenges of Hope Speech Detection in Code- Mixed Tulu Language shared task organized by DravidianLangTech@ACL 2026. To address the linguistic variability, orthographic inconsisten-

cies, and morphological richness of code-mixed social media content, we designed a multiple feature framework incorporating word n-grams, character n-grams, syllable n-grams, and sub-words representations. Ensemble of traditional ML classifiers with soft-voting demonstrated that integrating multiple feature representations significantly improves classification robustness and stability. Future directions include exploring class balancing techniques such as data augmentation and re-sampling to improve performance on minority classes. In addition, integrating transformer-based or multilingual contextual models, will be explored to enhance the system's ability to capture deeper semantic relationships in code-mixed text.

Acknowledgments

1. The authors are grateful to the Vision Group on Science and Technology (VGST), Department of Science and Technology, Government of Karnataka, for providing the financial support under the GRE scheme (GRD No: 1114) for carrying out this research work.

2. We acknowledge the use of generative AI systems in assisting with language refinement during manuscript preparation. All substantive contributions, including research design, experiments, and analysis, are carried out by the authors.

3. We thank Rachana N for her active participation and valuable contribution in this shared task. Her involvement and support played a meaningful role in the successful completion of this work.

Limitations

Despite its effectiveness, the proposed approach has several limitations.

- Shallow feature reliance: word, character, syllable n-grams and sub-words capture surface patterns but miss deeper semantic and contextual meaning. This limits performance on nuanced categories where subtle context matters.

- Sensitivity to noise: n-gram features are brittle in noisy, code-mixed, or transliterated text (common in Tulu social media). Spelling variations and inconsistent scripts reduce feature reliability.

- Curse of dimensionality: u-sing multiple n-gram and subword features creates very high-dimensional sparse vectors. Models like kNN and DT struggle with scalability and efficiency in such spaces.

- Limited generalization: Ensemble combinations (MNB+LR, kNN+DT) improve robustness but remain feature-based. They cannot generalize well to unseen contexts compared to contextual embeddings (e.g., BERT).

- Class imbalance vulnerability: probabilistic models like MNB and distance-based models like kNN are biased toward majority classes. Minority categories suffer disproportionately, even with ensemble voting.

- Overfitting risk: DT and kNN classifiers are prone to overfitting in small, imbalanced datasets. Without regularization or pruning, they memorize training patterns rather than learning general rules.

- Lack of semantic hierarchy: The models treat features independently and cannot capture hierarchical linguistic structures (syntax, discourse), which are often crucial in fine-grained hope speech detection.

The proposed methodology is strong enough for surface-level lexical cues, but limited in handling contextual semantics, noisy code-mixed text, and minority class generalization.

References

Nidhi Aggarwal, Ujjwal Singh, Venkatesh M. Reddy, and Aniket Puranik. 2023. Comparative Analysis of Machine Learning and BERT Models for Hope Speech Detection. *International Journal of Social Computing*.

Fazlourrahman Balouchzahi, Reza Farahbakhsh, and Noel Crespi. 2021a. Hope Speech Detection on Multilingual Social Media Texts Using Machine Learning, Deep Learning, and Transformer Models. In *Proceedings of the Workshop on Online Abuse and Harms*.

Fazlourrahman Balouchzahi, Reza Farahbakhsh, and Noel Crespi. 2021b. Using Ensemble Methods for Multilingual Hope Speech Detection. In *Proceedings of the Forum for Information Retrieval Evaluation (FIRE 2021) – Hope Speech Detection Shared Task*, pages –. ACM.

Bharathi Raja Chakravarthi. 2020. HopeEDI: A Multilingual Hope Speech Detection Dataset for Equality, Diversity, and Inclusion. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 247–256.

Asha Hegde, Bharathi Raja Chakravarthi, Hosahalli Lakshmaiah Shashirekha, Rahul Ponnusamy, Subalalitha Chinnaudayar Navaneethakrishnan, Lavanya Sambath Kumar, Durairaj Thenmozhi, Martha Karunakar, Shreya Sriram, and Sarah Aymen. 2023. Findings of the Shared Task on Sentiment Analysis in Tamil and Tulu Code-Mixed Text. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, pages 64–71, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Asha Hegde and Hosahalli Lakshmaiah Shashirekha. 2023. KT2: Kannada-Tulu Parallel Corpus Construction for Neural Machine Translation. In *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, pages 743–753, Goa University, Goa, India. NLP Association of India (NLPAI).

Aniket Puranik, Ujjwal Singh, and Venkatesh M. Reddy. 2023. Multilingual and Code-Mixed Hope Speech Detection using CNN and BiLSTM with BERT Variants. In *Proceedings of the IEEE International Conference on Natural Language Processing*.

A. Shahiki-Tash and 1 others. 2022. CNN-Based Hope Speech Detection for Spanish and English Social Media Text. *Journal of Language Technology and Applications*.

Poorvi Shetty. 2024. Natural Language Processing for Tulu: Challenges, Review and Future Scope. In *Proceedings of the International Conference on Speech and Language Technologies for Low-Resource Languages (SPELLL 2023)*, volume 2046 of *Communications in Computer and Information Science*, pages 93–109, Varna, Bulgaria. Springer Nature Switzerland.

Grigori Sidorov, Alexander Gelbukh, and Luis Chanona-Hernandez. 2021. Transformer-Based Models for Hope and Regret Speech Classification. In *International Conference on Computational Linguistics and Intelligent Text Processing*.

Tulio Ferreira Leite Da Silva, Gonzalo Freijedo Aduna, Farah Benamara, Alda Mari, Zongmin Li, Li Yue, and Jian Su. 2025. CDB: A Unified Framework for Hope Speech Detection through Counterfactual, Desire and Belief. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4448–4463, Albuquerque, New Mexico, USA. Association for Computational Linguistics.

Durairaj Thenmozhi, Rathnakar Shetty P, Parameshwar R. Hegde, Anusha M D, Raksha Adyanthaya, Mohammed Fadhel Aljunid, Prasanna Kumar Kumaresan, and Bharathi Raja Chakravarthi. 2026. Findings of the Shared Task on Hope Speech Detection in Tulu. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

NITC-HSR@DravidianLangTech 2026: Ensembling Multilingual Transformer Models for Detecting Abusive Tamil Text Targeting Women on Social Media

Rameez Mohammed A[1] **and S D Madhu Kumar**[2]
Department of Computer Science and Engineering
National Institute of Technology Calicut
Kozhikode, India
[1]rameez_p200051cs@nitc.ac.in, [2]madhu@nitc.ac.in

Abstract

The proliferation of misogynistic content on social media platforms is a serious problem that requires the development of automated detection systems, which is a challenging task for low-resource languages like Tamil. This study investigates the effectiveness of multilingual transformer models for identifying abusive Tamil text targeting women in social media. Results indicate that such models achieve strong baseline performance on this task. Furthermore, an ensemble of two best performing models was found to improve the classification performance further. The results also highlighted the significance of domain-specific pre-training for improving classifier performance. The best performing ensemble model achieved a weighted F1 score of 0.83 on the test set, placing our approach in first position in the shared task.

1 Introduction

Of late, social media has become an overwhelmingly accepted medium for dissemination of ideas, communication and entertainment. However, its widespread prevalence has also led to harmful behaviours like cyber bullying, hate speech, harassment etc. This has led to many adverse effects like psychological distress, marginalization of communities based on gender, religion, race etc. These challenges emphasize the need for developing robust detection and mitigation mechanisms to identify, prevent and reduce the harmful impacts of abusive and toxic content in social media environments.

Hate speech directed against women has become a pervasive issue in contemporary online spaces, reflecting deep-seated gender biases and discriminatory attitudes. Such expressions often manifest in various forms, including overt sexism, the use of vulgar or abusive language, derogatory remarks aimed at undermining women's opinions and achievements, and subtle forms of belittlement or stereotyping. The persistence of such content poses significant challenges to ensuring safe, inclusive, and respectful digital environments (Madhukumar and Mohammed A, 2023). Systems for detecting abusive content in social media have been widely researched and deployed for English, but there is a significant lack of resources and models for the same in regional Indian languages. As such, developing reliable mechanisms to identify abusive content targeted at women in regional Indian languages is a significant research problem.

A shared task of DravidianLangTech 2026 focuses on detecting abusive Tamil text aimed at women in social media, and the organizers have released a dataset of comments in Tamil towards the same. The dataset contains comments sourced from social media which are classified as either Abusive of Non-Abusive (Sivagnanam et al., 2026). A number of multilingual transformers like mBERT, IndicBERT etc., were fine-tuned on this dataset and their performance were evaluated. The effectiveness of AbuseXLMR (Gupta et al., 2022), a domain-adapted transformer model pre-trained on abusive speech corpora in several Indian languages, was also assessed on this dataset. Finally, experiments were conducted with an ensemble of the two best performing transformer models to investigate potential performance improvements through model combination. The python code for the experiments can be found in GitHub. [1]

2 Related Work

Several works for detecting hate and abusive speech have been carried out for English texts, but the suitability of these methods for regional Indian languages like Tamil is under-researched. This can largely be attributed to the paucity of high quality labeled datasets in Indian languages.

[1]https://github.com/rameezp200051cs/DravidianLangTech-2026

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 316–320
July 3, 2026 ©2026 Association for Computational Linguistics

Shared tasks associated with HASOC Chakravarthi et al., 2020, 2021, DravidianLangTech Priyadharshini et al., 2022, 2023; Premjith et al., 2024; Rajiakodi et al., 2025 etc. have provided datasets in multiple Indian languages for hate and abusive speech detection. These datasets have contributed significantly towards research related to hate speech detection mechanisms for monolingual, as well as code-mixed low resource Indian languages.

Saha et al. (2021) used multilingual transformers like XLM-RoBERTA, MuRIL and IndicBERT as well a BERT-CNN fusion model to classify code-mixed offensive comments in three Dravidian languages - Tamil, Kannada and Malayalam. They observed that a Genetic Algorithm based ensembling helped improve the performance of the models.

Subramanian et al. (2022) evaluated the performance of various ML models and transformer models on a dataset of Tamil Youtube comments and found that transformer models performed better in the task.

Gupta et al. (2022) released the MACD dataset, which is a large scale annotated dataset of abusive comments in five Indian languages. They further released AbuseXLMR, a pretrained abuse detection model based on XLM-RoBERTa model, for social media content in Indian languages, including Tamil.

Sreelakshmi et al. (2024) studied different multilingual transformer models with the objective of finding a single pre-trained embedding model which can be effective in detecting hateful and offensive language in code-mixed Kannada, Tamil and Malayalam.

Habiba and Aghila (2025) used an LSTM based approach for detecting abusive language in code-mixed Malayalam and Tamil languages as part of a DravidianLangTech 2025 shared task. Hanif et al. (Hanif and Rahman, 2025) compared the performance of multilingual transformer models for the same task and observed that IndicBERT gave the best results for both the languages.

3 Dataset Description

The datasets provided for the task contained Tamil language text sourced from social media. The texts were classified as either Abusive or Non-Abusive, depending on whether the text implied misogyny or not. Train and test datasets were made available separately by the organizers of the task. Both datasets are nearly balanced. The dataset statistics are as shown in Table 1.

Dataset	Abusive	Non-Abusive	Total
Train	1769	1883	3652
Test	441	472	913

Table 1: Dataset statistics

4 Preprocessing Data

An exploratory analysis of the text provided in the datasets indicated low code-mixing. Preprocessing of the text was done to remove URLs, hashtags, email addresses, and extra whitespaces. Hashtag symbols were also removed, but the associated keyword text were retained since it may contribute useful signals for classification.

5 Methodology

Multilingual transformer models have been found to be effective for offensive speech detection tasks in low-resource languages like Tamil. Five multilingual transformer models - XLM-RoBERTa (Conneau et al., 2020), mBERT (Devlin et al., 2019), MuRIL (Khanuja et al., 2021), IndicBERTv2 (Doddapaneni et al., 2023) and AbuseXLMR (Gupta et al., 2022) were used for the classification task. Of these models, XLM-RoBERTa and mBERT are pre-trained on a wide variety of languages spoken worldwide. MuRIL and IndicBERTv2 are pre-trained in Indian languages. AbuseXLMR is pre-trained exclusively on abusive social media comments in five Indian languages. The models have all been pre-trained in Tamil, which makes them suitable to be employed for the given task. All models were fine-tuned on 80% of the training dataset. 20% of the training set was designated as the validation set and the model performance was evaluated on the same. We fine-tuned models using the Hugging Face Transformers library with PyTorch. AdamW was used as the optimizer. The hyperparameters used for training the models are as shown in Table 2. They were obtained by

Hyperparameters	Values
Learning Rate	2e-5
Epochs	7
Weight Decay	0.01
Batch Size	16

Table 2: Hyperparameters used for training

Model	Accuracy	Precision	Recall	F1 Score
mBERT	0.7836	0.7438	0.8446	0.7910
XLM-RoBERTa	0.8098	0.7961	0.8164	0.8061
AbuseXLMR	0.8194	0.7761	0.8816	**0.8254**
MuRIL	0.8276	0.8295	0.8107	0.82
IndicBERTv2	0.8304	0.8267	0.8220	0.8244

Table 3: Performance of multilingual transformers on the train set

Ensembling Strategy	Accuracy	Precision	Recall	F1 Score
Soft Voting	0.8386	0.8172	0.8588	0.8375
Hard Voting	0.8249	0.8087	0.8362	0.82
Weighted Voting	0.8386	0.8172	0.8586	0.8375
Stacking - Logistic Regression	0.8497	0.8588	0.8249	**0.8415**
Stacking - Random Forest	0.83	0.8452	0.8022	0.8232

Table 4: Performance of AbuseXLMR - IndicBERTv2 ensemble on train set

manual tuning, following the commonly adopted settings for fine-tuning transformer models for text classification tasks.

5.1 Ensembling Models

Model ensembling is often employed as an effective strategy for improving classification performance, by leveraging the complementary strengths of the individual models. As such, we explored whether combining the predictions of the models under consideration could lead to further gains in classifier performance. As can be observed from Table 3, AbuseXLMR and IndicBERTv2 achieved the highest individual F1 scores. Accordingly, an ensemble of these two best performing models were evaluated with four different strategies:

- Soft voting - averages the predicted class probabilities of the models to determine the final classification
- Hard voting - takes a majority vote of the class labels predicted by the models to determine the final classification
- Weighted voting - combines predictions by giving higher weights to the predictions of the better performing model
- Stacking - trains meta-learners (linear regression, random forest) on the predictions of the models

6 Results

Table 3 reports the performance metrics obtained for the multilingual models on the classification task. The metrics listed are accuracy, precision, recall and macro F1 score. All the models perform strongly on the task, as evidenced by the metrics obtained. AbuseXLMR achieves the highest F1 score of 0.8254, followed by IndicBERTv2 with an F1 score of 0.8244. Since AbuseXLMR has been pretrained on abusive texts in Tamil (among other languages), it is able to perform marginally better than IndicBERTv2, which itself has been trained extensively on Indian languages. However, AbuseXLMR performs significantly better than its base model, XLM-RoBERTa, underscoring the fact that domain specific pretraining can lead to improvements over general-purpose multilingual models for abusive speech detection tasks.

Table 4 reports the performance of ensembled configurations of the two best performing multilingual models, AbuseXLMR and IndicBERTv2. Ensemble models exhibit stronger performance on the classification task as compared to the various multilingual models evaluated individually, as is evident from the higher values for the various metrics. Stacking with logistic regression strategy achieved the highest F1 score of 0.8415, combining the predictions from AbuseXLMR and IndicBERTv2 models. The better performance of this ensemble may be due to the complementary strengths of the two models - domain-specific pre-training (AbuseXLMR) and language-specific representations (IndicBERTv2), which the meta-learner combines adaptively based on prediction confidence and instance characteristics.

Table 5 presents the evaluation metrics achieved by the best performing ensemble model on the test

Model	Accuracy	Precision	Recall	Macro F1 Score	Weighted F1 Score
AbuseXLMR + IndicBERTv2 Ensemble (Stacking - Logistic Regression)	0.8302	0.8310	0.8293	0.8297	0.83

Table 5: Performance of AbuseXLMR - IndicBERTv2 ensemble on test set

set provided by the organizers. The model obtained a macro F1 of 0.8297 and weighted F1 of 0.83, placing our approach in first position in the shared task. The second placed team obtained a macro F1 of 0.8133 and weighted F1 of 0.81, whereas the third placed team obtained a macro F1 of 0.8103 and weighted F1 of 0.80.

7 Conclusion

This work investigates the use of multilingual transformer models for the task of detecting abusive Tamil text aimed at women in social media. Our experiments demonstrate that pre-trained multilingual models achieve strong baseline performance on this task. Moreover, it was observed that a domain-adapted model, AbuseXLMR, outperformed other models, highlighting the importance of task-specific pre-training. Ensembling AbuseXLMR and IndicBERTv2 models led to additional performance gains, highlighting the benefits of combining complementary model strengths. As future work, we plan to extend this approach for misogyny detection in other low-resource Indian languages as well.

8 Limitations

Dataset scarcity is the main hurdle to creating effective abuse detection models for low-resource languages like Tamil. The dataset provided for the shared task was used as is, with minimal preprocessing, for fine-tuning the models. Augmenting the dataset, like using back-translation, synonym replacements etc. may have improved the classification performance of the models. The models discussed in the paper are variants of multilingual transformers, fine tuned on the provided dataset. An evaluation of classifiers based on conventional machine learning models like SVM, Random Forest etc. was not conducted. A relative comparison of the performance of the classifiers based on these two approaches could have been informative.

9 GenAI Usage Disclosure

Anthropic's Claude was used exclusively for language refinement of the manuscript and debugging the code scripts associated with our methodology. No original text, arguments, or experimental conclusions were generated by the AI.

References

Bharathi Raja Chakravarthi, Prasanna Kumar Kumaresan, Ratnasingam Sakuntharaj, Anand Kumar Madasamy, Sajeetha Thavareesan, B Premjith, K Sreelakshmi, Subalalitha Chinnaudayar Navaneethakrishnan, John P McCrae, and Thomas Mandl. 2021. Overview of the hasoc-dravidiancodemix shared task on offensive language detection in tamil and malayalam. In *FIRE (Working Notes)*, pages 589–602.

Bharathi Raja Chakravarthi, Anand Kumar M, John P McCrae, Bhavukam Premjith, KP Soman, and Thomas Mandl. 2020. Overview of the track on hasoc-offensive language identification-dravidiancodemix. In *FIRE (Working notes)*, pages 112–120.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2023. Towards leaving no indic language behind: Building monolingual corpora, benchmark and models for indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12402–12426.

Vikram Gupta, Sumegh Roychowdhury, Mithun Das, Somnath Banerjee, Punyajoy Saha, Binny Mathew, Animesh Mukherjee, and 1 others. 2022. Multilingual abusive comment detection at scale for indic languages. *Advances in Neural Information Processing Systems*, 35:26176–26191.

A Habiba and G Aghila. 2025. Dltcnitpy@ dravidianlangtech 2025 abusive code-mixed text detection system targeting women for tamil and malayalam languages using deep learning technique. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 567–572.

Tareque Md Hanif and Md Rashadur Rahman. 2025. Cuet_agile@ dravidianlangtech 2025: Fine-tuning transformers for detecting abusive text targeting women from tamil and malayalam texts. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 315–319.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, and 1 others. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

SD Madhukumar and Rameez Mohammed A. 2023. Hate speech detection in indian languages: A brief survey. In *2023 IEEE 2nd International Conference on Data, Decision and Systems (ICDDS)*, pages 1–5. IEEE.

B Premjith, Bharathi Raja Chakravarthi, Prasanna Kumar Kumaresan, Saranya Rajiakodi, Sai Karnati, Sai Mangamuru, and Chandu Janakiram. 2024. Findings of the shared task on hate and offensive language detection in telugu codemixed text (hold-telugu)@ dravidianlangtech 2024. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 49–55.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, Subalalitha Cn, Kogilavani SV, Abirami Murugappan, Prasanna Kumar Kumaresan, and 1 others. 2023. Overview of shared-task on abusive comment detection in tamil and telugu. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, pages 80–87.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, Subalalitha Chinnaudayar Navaneethakrishnan, Thenmozhi Durairaj, Malliga Subramanian, Kogilavani Shanmugavadivel, Siddhanth U Hegde, and Prasanna Kumar Kumaresan. 2022. Findings of the shared task on abusive comment detection in tamil. In *Proceedings of the second workshop on speech and language technologies for Dravidian languages. Association for Computational Linguistics*.

Saranya Rajiakodi, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, Ruba Priyadharshini, Kathiravan Pannerselvam, Rahul Ponnusamy, Bhuvaneswari Sivagnanam, Paul Buitelaar, Jananayagan Jananayagan, Kishore Kumar Ponnusamy, and 1 others. 2025. Findings of the shared task on abusive tamil and malayalam text targeting women on social media: Dravidianlangtech@ naacl 2025. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 671–681.

Debjoy Saha, Naman Paharia, Debajit Chakraborty, Punyajoy Saha, and Animesh Mukherjee. 2021. Hate-alert@ dravidianlangtech-eacl2021: Ensembling strategies for transformer-based offensive language detection. In *Proceedings of the first workshop on speech and language technologies for Dravidian languages*, pages 270–276.

Bhuvaneswari Sivagnanam, Kathiravan Pannerselvam, Jananayagan V, Charmathi Rajkumar, Ramesh Kannan R, Ratnavel Rajalakshmi, Shunmuga Priya Muthusamy Chinnan, Saranya Rajiakodi, and Bharathi Raja Chakravarthi. 2026. From Comments to Harm: A Findings Report on Abusive Tamil Text Targeting Women on Social Media. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

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

Malliga Subramanian, Rahul Ponnusamy, Sean Benhur, Kogilavani Shanmugavadivel, Adhithiya Ganesan, Deepti Ravi, Gowtham Krishnan Shanmugasundaram, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. 2022. Offensive language detection in tamil youtube comments by adapters and cross-domain knowledge transfer. *Computer Speech & Language*, 76:101404.

PhucNguyen@DravidianLangTech 2026: Political Multiclass Sentiment Analysis with XLM-RoBERTa and Low-Rank Adaptation

Dinh Khac Phuc Nguyen[1,2] **and Dang Van Thin**[1,2]
[1]University of Information Technology, VNU-HCM, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
24521172@gm.uit.edu.vn, thindv@uit.edu.vn

Abstract

Analyzing political sentiment in code-mixed Tamil-English presents significant challenges due to informal jargon, severe class imbalance, and distribution shifts. This paper describes our system for the Political Multiclass Sentiment Analysis shared task at DravidianLangTech@ACL 2026, which categorizes tweets into seven sentiment classes. Our approach leverages XLM-RoBERTa integrated with Low-Rank Adaptation (LoRA). To mitigate majority-class dominance, we combine random oversampling with automated hyperparameter optimization to improve macro-level balance within this Parameter-Efficient Fine-Tuning (PEFT) framework. Enhanced by targeted preprocessing—specifically emoji demojization and noise removal—our system helps preserve nuanced symbolic cues, achieving a macro-average F1-score of 0.3763 and securing Rank 2 on the shared task leaderboard.

Keywords: Sentiment Analysis, Code-Mixed Data, Tamil tweets, XLM-RoBERTa, PEFT Tuning.

1 Introduction

Analyzing political sentiment in low-resource, code-mixed languages like Tamil-English is highly challenging due to noisy user-generated content and severe class imbalance, which frequently causes neural models to exhibit systematic majority-class bias (Chakravarthi et al., 2020). While multilingual models such as XLM-R (Conneau et al., 2019) and Parameter-Efficient Fine-Tuning (PEFT) methods like LoRA (Hu et al., 2022) offer robust and efficient solutions, their behavior under extreme class skew remains empirically under-characterized in shared-task settings. Specifically, we observe that default learning rate configurations in LoRA can rapidly exacerbate majority-class dominance, resulting in severe prediction bias toward dominant categories.

In this work, we investigate LoRA-based fine-tuning for Political Multiclass Sentiment Analysis of Tamil-English tweets (Vegupatti et al., 2026). By coupling random oversampling of the training data with a systematic hyperparameter search via Optuna, we demonstrate that exploring learning rate and weight decay sensitivity effectively mitigates majority-class over-indexing and improves macro-level balance. Furthermore, we show that domain-specific preprocessing—namely emoji demojization and hashtag normalization—helps preserve essential symbolic and lexical cues in informal text.

Our contributions are threefold:

- We detail an optimized LoRA adaptation pipeline under severe class imbalance, highlighting the model's sensitivity to hyperparameter configurations and the practical necessity of automated tuning.
- We demonstrate that targeted preprocessing (demojization and hashtag segmentation) effectively retains semantic signals in code-mixed political discourse.
- Our system achieves a macro-F1 score of 0.3763—an absolute improvement of +0.0331 over a standard full fine-tuning baseline—securing Rank 2 in the DravidianLangTech 2026 shared task (Vegupatti et al., 2026).

The remainder of this paper is organized as follows. Section 2 reviews related work relevant to our study. Section 3 presents the task and dataset description, including the problem formulation and data characteristics. Section 4 details the proposed methodology. Section 5 describes the experimental setup and reports the results. Section 6 provides discussion and in-depth analysis of the findings. Finally, Section 7 concludes the paper and outlines directions for future work.

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 321–325
July 3, 2026 ©2026 Association for Computational Linguistics

2 Related Work

Code-Mixed Sentiment Analysis. Sentiment analysis in Tamil-English (Tanglish) is hindered by informal grammar, phonetic typing (Chakravarthi et al., 2020; Hande et al., 2021), and symbolic noise (e.g., emojis), which standard subword tokenizers inconsistently segment (Priyadharshini et al., 2021). Furthermore, Dravidian political sentiment datasets exhibit severe skew toward opinionated content, exacerbating minority class sparsity. This necessitates domain-specific text normalization to preserve contextual cues prior to encoding.

Multilingual Models and PEFT. While multilingual models like XLM-RoBERTa (Conneau et al., 2019) excel in cross-lingual transfer, full fine-tuning on small, skewed datasets frequently causes unstable adaptation and overfitting. Parameter-Efficient Fine-Tuning (PEFT) methods like LoRA (Hu et al., 2022) mitigate this by restricting the trainable parameter space. However, under extreme class imbalance, this reduced capacity may amplify sensitivity to hyperparameter configurations, leaving PEFT optimization behavior less explored in shared-task contexts.

Optimization under Class Imbalance. Traditional imbalance mitigations like Focal Loss (Lin et al., 2017) or data resampling can introduce instability or increase overfitting risks under noisy supervision. To manage these trade-offs, automated hyperparameter optimization (Akiba et al., 2019) is utilized. Unlike prior work focusing solely on objective re-weighting, we emphasize the interaction between data-level and optimization-level interventions. We empirically explore how coupling data resampling with automated tuning within the constrained LoRA space improves training stability and mitigates majority-class dominance, offering a practical alternative to bespoke loss engineering.

3 Task and Dataset Description

3.1 Task Definition and Evaluation

The DravidianLangTech@ACL 2026 shared task (Vegupatti et al., 2026) requires classifying code-mixed Tamil tweets into seven political sentiment classes. Systems are evaluated using the macro-averaged F1-score. By assigning equal weight to each class, this metric strictly emphasizes minority-class performance, making robustness to imbalance crucial.

Table 1: Class distribution of the official datasets. The severe skew toward *Opinionated* presents a significant optimization challenge.

Sentiment Class	Train (%)	Dev (%)
Opinionated	1,361 (31.3)	153 (28.1)
Sarcastic	790 (18.2)	115 (21.1)
Neutral	637 (14.6)	84 (15.4)
Positive	575 (13.2)	69 (12.7)
Substantiated	412 (9.5)	52 (9.6)
Negative	406 (9.3)	51 (9.4)
None of the above	171 (3.9)	20 (3.7)
Total	**4,352 (100)**	**544 (100)**

3.2 Dataset Statistics and Optimization Challenges

The Tanglish dataset includes 4,352 training and 544 development instances, featuring informal text rich in political hashtags and emojis. The class distribution remains consistent across both splits.

As shown in Table 1, the data is severely skewed: *Opinionated* dominates (31.3%), while *Substantiated* and *Negative* fall below 10%. In practice, full fine-tuning under such imbalance often yields majority-class-biased predictions, as frequent labels dominate the optimization objective. We hypothesize that LoRA's restricted trainable parameter space influences adaptation dynamics under skewed supervision, potentially moderating abrupt majority-class bias compared to full-parameter updates.

4 Methodology

Our system integrates contextual text normalization, cross-lingual encoding, and parameter-efficient optimization to address the vulnerabilities of pretrained models under extreme class skew and phonetic noise.

4.1 Contextual Preprocessing

Political discourse relies heavily on symbolic cues frequently inconsistently segmented by subword tokenizers. We implemented targeted preprocessing to anchor these within the embedding space: (1) **Emoji Demojization** via the `emoji` library to translate symbols into descriptive text, and (2) **Structural Noise Removal** filtering URLs and user mentions. Additionally, sequences were padded and truncated to a maximum length of 256 tokens to balance contextual coverage and computational efficiency.

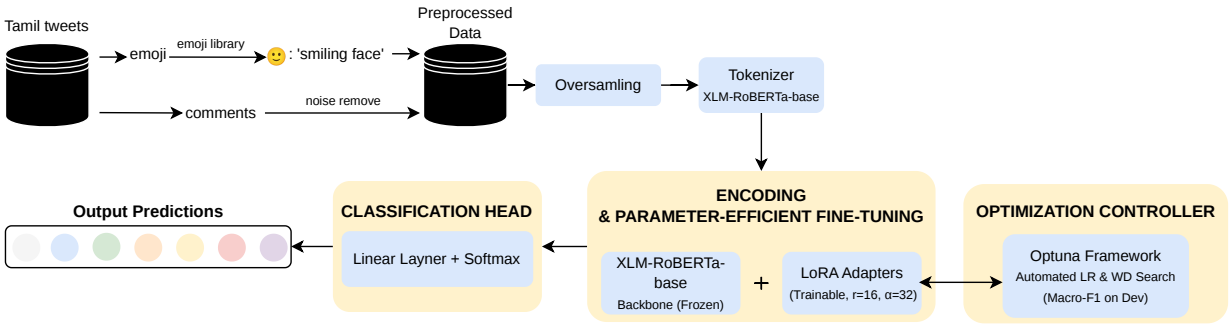


Figure 1: Overview of the XLM-R + LoRA pipeline with Optuna-based hyperparameter optimization for robust training under class imbalance and noisy social media text.

4.2 Multilingual Foundation Model

We adopted **XLM-RoBERTa-base** (Conneau et al., 2019), leveraging its robust cross-lingual representations for Tanglish text without requiring external translation pipelines.

4.3 Parameter-Efficient Adaptation via LoRA

To prevent severe overfitting on small, skewed datasets, we applied **Low-Rank Adaptation (LoRA)** (Hu et al., 2022). LoRA freezes pre-trained weights and introduces trainable low-rank matrices into the self-attention layers. We targeted the *query* and *value* projections ($r = 16$, $\alpha = 32$, dropout=0.05), reducing trainable parameters to under 1%, thereby limiting adaptation capacity and improving parameter efficiency. All configurations utilized mixed-precision (FP16) training to optimize memory utilization.

4.4 Imbalance Mitigation via Optimization

To alleviate initial data sparsity, we applied random oversampling (Lemaître et al., 2017) exclusively to the training set. However, naive oversampling risks overfitting to duplicated minority-class noise if the learning process is not carefully controlled.

To systematically navigate the constrained LoRA parameter space, we utilized **Optuna** (Akiba et al., 2019) for hyperparameter search. Empirical tuning identified an optimal hyperparameter configuration—specifically a learning rate of 1.31×10^{-4} (AdamW) and a weight decay of 0.098—providing stable adaptation. This hybrid strategy effectively improves minority-class performance without the need for bespoke loss engineering.

Table 2: Hyperparameter optimization details explored via Optuna.

Hyperparameter	**Search Space**	**Best Value**
Learning Rate	$[1 \times 10^{-4}, 5 \times 10^{-4}]$	1.31×10^{-4}
Weight Decay	$[0.01, 0.1]$	0.098
Num. Train Epochs	$[6, 15]$	10
AdamW β_1	$[0.8, 0.99]$	0.8
AdamW ϵ	$[10^{-8}, 5 \times 10^{-6}]$	6×10^{-8}
Fixed Parameters		
LoRA Rank (r)	16	16
LoRA Alpha (α)	32	32
Batch Size	32	32

Table 3: System performance on the test set.

Model Configuration	**Accuracy**	**Macro F1**
XLM-R (Full Fine-Tuning)	0.3033	0.3432
XLM-R + LoRA (Standard)	0.3033	0.3193
XLM-R + LoRA + Focal Loss	0.3511	0.3286
XLM-R + LoRA + Optuna (Ours)	**0.3364**	**0.3763**

5 Experimental Setup and Results

5.1 Evaluation Metrics

Following shared task guidelines, we evaluate system performance using the macro-averaged F1-score to ensure smaller sentiment categories are appropriately weighted. This prevents the predominant *Opinionated* class from disproportionately skewing the perceived efficacy. Overall accuracy is also reported for completeness.

5.2 Comparative Results

Table 3 details the performance across configurations. Our final pipeline—integrating XLM-RoBERTa, LoRA, and Optuna—achieved a macro-F1 of 0.3763 and an accuracy of 0.3364 on the official blind test set, securing 2nd place in the shared task.

5.3 Analysis

The results highlight the complexities of adaptation under extreme class skew. Standard LoRA underperforms the full fine-tuning baseline in macro-F1 (0.3193 vs. 0.3432). Notably, while Focal Loss achieves the highest overall accuracy (0.3511), its lower macro-F1 (0.3286) suggests less balanced performance across sentiment categories, as evidenced by the discrepancy between overall accuracy and macro-averaged metrics.

Conversely, systematic hyperparameter search via Optuna yielded a notable improvement of **0.0331 macro-F1 points** over the baseline. The contrast between Focal Loss and Optuna-based tuning highlights an important trade-off between objective-level re-weighting and optimization-level control. While Focal Loss explicitly reshapes gradient contributions, its performance remains highly sensitive to static hyperparameter choices. In contrast, automated search enables adaptive calibration of optimization behavior within the constrained LoRA space, reducing performance volatility across configurations. This suggests that, under extreme imbalance, tuning the training dynamics itself is as critical as modifying the loss formulation.

6 Discussion and Analysis

6.1 Error Analysis and Confusion Patterns

To evaluate qualitative performance, we analyzed the confusion matrix generated from the official test set (Figure 2). The analysis reveals two prominent sources of confusion:

- **Neutral vs. Opinionated Ambiguity:** The model frequently misidentifies *Neutral* reports as *Opinionated* due to high lexical overlap. Distinguishing objective reporting from subtle political bias remains challenging without broader pragmatic cues.
- **Sarcasm Recognition Bottlenecks:** *Sarcastic* instances are often confounded with *Opinionated* and *Neutral* labels. This potentially stems from complex phonetic wordplay in code-mixed Tanglish which current preprocessing cannot explicitly normalize, leading to sub-optimal representation.

Despite these bottlenecks, the *Substantiated* class maintains non-trivial recall under severe data sparsity. This indicates a partial retention of minority-class signals, demonstrating that the optimization regime does not entirely suppress underrepresented categories.

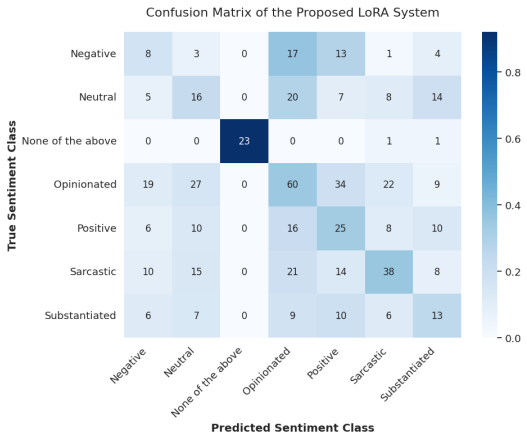


Figure 2: Confusion matrix of the XLM-R + LoRA system.

6.2 Optimization in Bias Mitigation

Compared to standard full fine-tuning, the constrained LoRA parameterization combined with automated hyperparameter search resulted in more balanced confusion patterns. This suggests that identifying a specific hyperparameter configuration empirically correlates with improved minority-class performance, yielding more balanced predictions across sentiment categories under skewed supervision.

7 Conclusion

This paper presented a high-performing pipeline for Political Multiclass Sentiment Analysis of code-mixed Tamil comments. By integrating emoji-aware preprocessing with Low-Rank Adaptation and random oversampling, our system achieved Rank 2 (macro-F1: 0.3763). While this absolute score underscores the inherent complexity of the code-mixed Dravidian linguistic landscape, our results suggest that algorithmically optimizing the adaptation process via Optuna provides a practical complement to static interventions, improving macro-level balance in low-resource, severely imbalanced settings without bespoke loss engineering.

Although our study focuses on code-mixed Tamil political discourse, the observed optimization dynamics may extend to other low-resource multilingual classification settings characterized by heavy class skew. These findings motivate further investigation into the interaction between parameter-efficient adaptation and automated hy-

perparameter search in broader cross-lingual contexts.

Acknowledgements

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

Limitations

While our pipeline achieved a competitive Rank 2, the absolute macro-F1 score of 0.3763 highlights the severe difficulty of the code-mixed Tanglish domain. A primary limitation is the persistent confounding of *Sarcastic*, *Neutral*, and *Opinionated* classes. We hypothesize this stems from complex phonetic wordplay and high lexical overlap, which standard tokenizers and our current preprocessing cannot fully resolve. Furthermore, our methodology relies on integrating existing optimization frameworks rather than proposing novel architectures. Addressing these systemic ambiguities will require exploring bespoke loss functions and architectural designs explicitly tailored for phonetic variations in low-resource settings.

Ethical Considerations

This research strictly utilizes the publicly available dataset provided by the DravidianLangTech@ACL 2026 organizers. We do not attempt to deanonymize users or extract personally identifiable information. The models developed are intended for academic research in computational linguistics and should not be deployed for automated moderation without further fairness and bias evaluations.

AI Usage Acknowledgment

Following the ACL 2026 guidelines, we acknowledge the use of Large Language Models (LLMs) solely for the purposes of English proofreading and LaTeX formatting assistance during the preparation of this manuscript. The authors assume full responsibility for the final content and findings.

Reproducibility

The source code, hyperparameter search scripts, and training configurations are publicly available at: https://github.com/nguyenbbt/ACL2026_DravidianLangTech

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.

Bharathi Raja Chakravarthi, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John P. McCrae. 2020. Corpus creation for sentiment analysis in code-mixed Tamil-English text. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 202–210. European Language Resources Association.

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. *arXiv preprint arXiv:1911.02116*.

Adeep Hande, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. 2021. Offensive language identification in low-resourced code-mixed Dravidian languages using Multilingual Transformers. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 104–110.

Edward J Hu, Yelong Shen, Phil Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Guillaume Lemaître, Fernando Nogueira, and Christos K Aridas. 2017. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, 18(1):559–563.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, S. Rajalakshmi, and Premjith B. 2021. Named entity recognition for code-mixed Indian languages using Transformer models. In *Proceedings of the 1st Workshop on Speech and Vision for Dravidian Languages*, pages 1–7.

Mani Vegupatti, Kishore Kumar Ponnusamy, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Durairaj Thenmozhi, Prasanna Kumar Kumaresan, and Sathiyaraj Thangasamy. 2026. TamilPoliSent 2026: A Shared Task report on Multiclass Political Sentiment Analysis in Tamil. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

PolyTicsTamil_Alchemists@DravidianLangTech@ACL 2026: An Augmentation-Driven Focal Ensemble Model for Political Sentiment Analysis in Tamil

Jyoti Kumari[1] **Meclin A Francis**[2] **Vinay Babu Ulli**[3]
Malavika Sreekumar[4] **Joel Johnson**[5]
[1]Department of Linguistics, Banaras Hindu University, Varanasi, India
[2]Amrita School of Artificial Intelligence, Coimbatore, Amrita Vishwa Vidyapeetham, India
[3]Oogwai Analytics, Bangalore, India
[4]TransUnion, Pune, India [5]IBM, Kochi, India
meclinafrancis@gmail.com

Abstract

This paper describes our system submitted to the DravidianLangTech@ACL 2026 shared task on Political Multiclass Sentiment Analysis of Tamil X (Twitter) Comments. The task requires classifying Tamil political tweets into seven sentiment categories. We address two key challenges, severe class imbalance and semantic overlap between categories, through a three-stage pipeline. First, we balance the training set by augmenting minority classes via back-translation and transformer-based paraphrasing. Second, we fine-tune XLM-RoBERTa-base using a class-weighted Focal Loss (γ=2), which directs learning towards hard, ambiguous samples. Third, we train five models under Stratified 5-Fold Cross-Validation and average their softmax outputs at inference time. On the official test set, the system achieves a Macro-F1 of **0.3539**. The code is publicly available.[1]

1 Introduction

Social media platforms, X (formerly Twitter) in particular, have emerged as primary channels for political discourse. Automatically analysing the sentiment of such content can inform political communication research and public opinion tracking. However, political text is challenging for sentiment analysis due to rhetorical devices, sarcasm, and culturally coded references.

These difficulties are compounded for Tamil, which is morphologically rich and agglutinative. Political commentary on Tamil social media further exhibits pervasive code-mixing with English, non-standard orthography, and dialectal variation.

The DravidianLangTech@ACL 2026 shared task (Vegupatti et al., 2026) on *Political Multiclass Sentiment Analysis of Tamil X (Twitter) Comments* formalises this as a seven-way classification task with labels: *Negative*, *Neutral*, *None of the above*, *Opinionated*, *Positive*, *Sarcastic*, and *Substantiated*. Two aspects make this task particularly difficult: the label distribution is heavily skewed, and certain label pairs such as *Opinionated* versus *Sarcastic* share overlapping surface features.

We address these challenges through three stages: (1) synthetic augmentation of underrepresented classes using back-translation and paraphrasing until all classes reach equal frequency; (2) fine-tuning XLM-RoBERTa-base (Conneau et al., 2020) with class-weighted Focal Loss (Lin et al., 2017); and (3) ensembling five models trained under Stratified 5-Fold Cross-Validation via probability averaging.

2 Related Work

The DravidianLangTech workshop series has highlighted the difficulties of processing Tamil social media text (Chakravarthi et al., 2020). The preceding DravidianLangTech@NAACL 2025 iteration introduced this political sentiment task with the same seven-class label set (Chakravarthi et al., 2025), where overall Macro-F1 scores remained modest despite 25 participating teams. The top-ranked Synapse team combined IndicBERTv2-MLM features with TF-IDF and back-translation (Kp et al., 2025). Roy et al. (2025) fine-tuned L3Cube-Tamil-BERT, while Shanmugavadivel et al. (2025) paired Random Forest with feedforward networks. S et al. (2025) employed SGD with incremental learning, and K et al. (2025) embedded tweets with LaBSE and classified with SVMs.

Beyond this shared-task series, processing low-resource Indian languages like Tamil has seen growing interest, with researchers frequently grappling with code-mixing and morphological complexity. Furthermore, handling multi-class imbalance in text classification is a widely recognized challenge in the broader NLP literature. Researchers frequently employ oversampling, cost-sensitive learn-

[1]https://github.com/meclin2345/PolyTicsTamil_Alchemists

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 326–330
July 3, 2026 ©2026 Association for Computational Linguistics

ing, and advanced objective functions to prevent majority class dominance.

A common limitation is that class imbalance is typically addressed at only one level. Our work departs by enforcing class parity through augmentation, pairing it with Focal Loss for hard-example mining, and aggregating predictions across five folds to mitigate overfitting.

3 Methodology

We frame the task as a multi-class classification problem over seven political sentiment categories. Our system consists of three stages: data augmentation to address class imbalance, fine-tuning of a pre-trained transformer encoder, and cross-validation ensembling with focal loss.

3.1 Data Augmentation

Let N_c denote the number of training samples in class c, and $N_{\max} = \max_c N_c$. For every class where $N_c < N_{\max}$, we generate $N_{\max} - N_c$ additional samples using two complementary techniques. Back-translation translates each Tamil tweet to English and back to Tamil using Google Translate, producing structural paraphrases. Transformer-based paraphrasing uses IndicBART (Dabre et al., 2022) to generate lexical and syntactic variants. We alternate between the two techniques until all classes reach $N_{\max}$ samples, yielding a uniform training distribution.

3.2 Model Architecture and Loss Function

We use XLM-RoBERTa-base (Conneau et al., 2020) as our encoder. The `[CLS]` token representation $\mathbf{z} \in \mathbb{R}^{768}$ is passed through a dropout layer (p=0.4) and a linear projection to produce logits $\mathbf{o} \in \mathbb{R}^7$.

We train with class-weighted Focal Loss (Lin et al., 2017), which introduces a modulating factor $(1-p_t)^\gamma$ that suppresses the loss for well-classified examples:

$$\mathcal{L}_{\text{FL}} = -\alpha_c \, (1 - p_t)^\gamma \, \log(p_t) \qquad (1)$$

where p_t is the predicted probability for the ground-truth class, γ=2 is the focusing parameter, and $\alpha_c = N_{\text{total}}/(C \cdot N_c)$ is the per-class weight.

3.3 Stratified k-Fold Ensemble

We employ Stratified 5-Fold Cross-Validation to reduce variance from any single split. Five independent XLM-R models are trained, each using four folds for training and one for validation. At inference, all five models produce softmax vectors and the final prediction is obtained by averaging:

$$\hat{y} = \arg\max_j \; \frac{1}{K} \sum_{i=1}^{K} \text{softmax}(\text{Model}_i(x))_j \qquad (2)$$

where K=5. This soft-voting strategy preserves confidence information, allowing highly certain models to exert proportionally greater influence.

Category	Original	After Aug.
Opinionated	1,361	1,361
Sarcastic	790	1,361
Neutral	637	1,361
Positive	575	1,361
Substantiated	412	1,361
Negative	406	1,361
None of the above	171	1,361
Total	4,352	9,527

Table 1: Class distribution before and after augmentation.

Hyperparameter	Value
Pre-trained model	`xlm-roberta-base`
Optimizer	AdamW
Learning rate	1×10^{-5}
LR scheduler	Linear with warm-up
Batch size	16
Epochs per fold	5
Max sequence length	128
Dropout rate	0.4
Focal Loss γ	2.0
Number of folds (K)	5

Table 2: Hyperparameter configuration.

4 Experimental Setup

4.1 Dataset

Table 1 reports the original class distribution. After augmentation, minority classes are expanded to match the majority class count ($N_{\max}$=1,361). The test set contains 544 unlabeled samples.

4.2 Implementation Details

Our pipeline was implemented in PyTorch using the Hugging Face `transformers` library (Wolf et al., 2020) on a single NVIDIA GPU. Table 2 summarises the hyperparameters. For each fold, we saved the checkpoint with the highest validation Macro-F1.

Configuration	Val F1
XLM-R, no aug., CE loss	0.2200
XLM-R, aug., Focal Loss	0.4268
5-Fold Ensemble, aug., FL	**0.4050**

Table 3: Ablation results. CE = Cross-Entropy, FL = Focal Loss. The single-split augmented model reports best validation F1; the ensemble reports mean CV F1.

Fold	Val Macro-F1
1	0.4158
2	0.3900
3	0.4038
4	0.4083
5	0.4069
Mean	**0.4050**

Table 4: Per-fold validation Macro-F1 scores.

Metric	Macro	Weighted
Accuracy	0.3474	
Precision	0.3493	0.3304
Recall	0.3722	0.3474
F1-Score	**0.3539**	0.3328

Table 5: Official test set results.

Class	P	R	F1
Negative	0.27	0.29	0.28
Neutral	0.52	0.12	0.20
None of the above	0.94	0.88	0.91
Opinionated	0.34	0.45	0.39
Positive	0.31	0.48	0.38
Sarcastic	0.54	0.48	0.51
Substantiated	0.37	0.30	0.33
Macro Avg	0.47	0.43	**0.43**

Table 6: Class-wise precision (P), recall (R), and F1 for the best single-split model on the validation set.

5 Results and Analysis

5.1 Validation Results

Table 3 presents an ablation comparing three configurations on the validation set. The baseline XLM-R model trained with standard cross-entropy on the original imbalanced data achieves a Macro-F1 of 0.2200. Adding augmentation and Focal Loss improves this to 0.4268, confirming that addressing imbalance at both data and loss-function levels yields complementary gains. The 5-Fold ensemble produces stable per-fold F1 scores (Table 4), with a mean cross-validation Macro-F1 of 0.4050.

Interestingly, the ensemble's mean CV Macro-F1 (0.4050) is lower than the best single-split model (0.4268). This suggests that our simple soft-voting strategy may occasionally over-average predictions, diluting highly confident correct outputs from individual folds.

5.2 Official Test Results

Table 5 reports the official test set results. The system achieves a Macro-F1 of **0.3539** with an accuracy of 0.3474. The drop from validation Macro-F1 (0.4268 single-split; 0.4050 mean CV) to test Macro-F1 (0.3539) indicates that the augmented training distribution does not fully capture the diversity of the test set. The gap between Macro-F1 (0.3539) and weighted F1 (0.3328) further suggests that the system performs relatively better on minority classes than on majority classes, consistent with the augmentation and Focal Loss strategy that explicitly upweights underrepresented categories.

5.3 Class-wise Analysis

Table 6 reports per-class performance from the best single-split model on the validation set. The system performs strongly on *None of the above* (F1 = 0.91), which is the most lexically distinct category. Performance on *Sarcastic* (0.51) is moderate, while *Neutral* (0.20) and *Negative* (0.28) remain challenging due to overlapping surface features with other categories.

5.4 Error Analysis

Three dominant error patterns emerge from manual inspection. First, **sarcasm–opinion confusion**: sarcastic tweets often adopt the surface structure of genuine opinions, differing only in pragmatic intent, leading to frequent misclassification between *Sarcastic* and *Opinionated*. Second, **low recall for Neutral**: the model struggles to identify neutral comments, as they lack the strong lexical signals that characterise opinionated or sarcastic text, resulting in many neutral samples being absorbed into adjacent categories. Third, **augmentation noise**: we did not formally assess the semantic validity of every generated sample. Manual inspection indicates that while back-translation and paraphrasing successfully increase class frequencies, they occasionally introduce translationese artifacts or alter the pragmatic meaning of tweets due to the uneven quality of machine translation for Tamil. This noise strongly contributes to the test-set generalization gap.

6 Conclusion

We presented our system for the DravidianLangTech@ACL 2026 shared task on Political Multiclass Sentiment Analysis of Tamil Twitter Comments (Vegupatti et al., 2026). Our three-stage pipeline addresses class imbalance through back-translation and paraphrasing-based augmentation, class-weighted Focal Loss (γ=2), and a Stratified 5-Fold ensemble. The ablation confirms that each stage contributes incremental gains, with the augmentation and Focal Loss combination yielding the largest improvement (from 0.2200 to 0.4268 validation Macro-F1). On the official test set, the system achieves a Macro-F1 of 0.3539. The gap between validation and test performance suggests that synthetic augmentation, while effective for balancing class frequencies, does not fully substitute for natural data diversity. Future work will explore incorporating discourse-level features to address sarcasm–opinion confusion, learned ensemble weights via a meta-classifier, and larger Indic-specific pre-trained models such as MuRIL as the backbone encoder.

Limitations

Our work has several limitations. First, the augmentation strategy relies on back-translation through English via Google Translate and paraphrasing with IndicBART, both of which may introduce translationese artefacts or semantic drift that do not reflect the natural distribution of Tamil political discourse; the gap between validation Macro-F1 (0.4050) and test Macro-F1 (0.3539) supports this concern. Second, we use only XLM-RoBERTa-base as our encoder and do not compare against Indic-specialised models such as MuRIL or IndicBERT, which may capture Tamil morphological patterns more effectively. Third, the seven sentiment categories exhibit inherent semantic overlap—particularly between *Sarcastic* and *Opinionated* that surface-level token representations struggle to disambiguate, yet we do not incorporate any discourse-level, pragmatic, or contextual features beyond the individual tweet. Fourth, the Focal Loss hyperparameter γ=2 and the dropout rate of 0.4 were selected based on common defaults rather than through systematic tuning, and may not be optimal for this label distribution. Fifth, our error analysis is qualitative and based on manual inspection of a small sample; we do not provide a formal confusion matrix on the test set or a quantitative breakdown of error types. Finally, the system is evaluated on a single dataset from one shared task; its ability to generalise to other Tamil sentiment benchmarks, different time periods, or other Dravidian languages remains untested.

References

Bharathi Raja Chakravarthi, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John P. McCrae. 2020. A sentiment analysis dataset for code-mixed Malayalam-English. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 177–184, Marseille, France. European Language Resources association.

Bharathi Raja Chakravarthi, Saranya Rajiakodi, Thenmozhi Durairaj, Sathiyaraj Thangasamy, Ratnasingam Sakuntharaj, Prasanna Kumar Kumaresan, Kishore Kumar Ponnusamy, Arunaggiri Pandian Karunanidhi, and Rohan R. 2025. Overview 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*, pages 746–753, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. 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. 2020. 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.

Raj Dabre, Himani Shrotriya, Anoop Kunchukuttan, Ratish Puduppully, Mitesh Khapra, and Pratyush Kumar. 2022. IndicBART: A pre-trained model for indic natural language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1849–1863, Dublin, Ireland. Association for Computational Linguistics.

Nithish Ariyha K, Eshwanth Karti T R, Yeshwanth Balaji A P, Vikash J, and Sachin Kumar S. 2025. Wictory@DravidianLangTech 2025: Political sentiment analysis of Tamil X(Twitter) comments using LaBSE and SVM. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 706–710, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Suriya Kp, Durai Singh K, Vishal A S, Kishor S, and Sachin Kumar S. 2025. Synapse@DravidianLangTech 2025: Multiclass 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.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007.

Billodal Roy, Pranav Gupta, Souvik Bhattacharyya, and Niranjan Kumar M. 2025. LexiLogic@DravidianLangTech 2025: Multimodal hate speech detection in Dravidian languages. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 552–556, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Kalaivani K S, Sanjay R, Thissyakkanna S M, and Nirenjhanram S K. 2025. KSK@DravidianLangTech 2025: Political multiclass sentiment analysis of Tamil X (Twitter) comments using incremental learning. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 221–225, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Kogilavani Shanmugavadivel, Malliga Subramanian, Sanjai R, Mohammed Sameer, and Motheeswaran K. 2025. Beyond_Tech@DravidianLangTech 2025: Political multiclass sentiment analysis using machine learning and neural network. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 139–143, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Mani Vegupatti, Kishore Kumar Ponnusamy, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Durairaj Thenmozhi, Prasanna Kumar Kumaresan, and Sathiyaraj Thangasamy. 2026. TamilPoliSent 2026: A Shared Task report on Multiclass Political Sentiment Analysis in Tamil. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 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.

PrimeLine@DravidianLangTech 2026: Abusive Tamil Comment Detection Using MuRIL

Rithikaa V Sanjay Krishnan K Nithya Varshini C N R S. Sumathi
Department of Information Technology, St. Joseph's College of Engineering
https://github.com/Nithya-svg/primeline_tamil_abusive

Abstract

Detecting abusive language in Tamil social media is a genuinely difficult problem. The language is morphologically rich, speakers routinely mix Tamil with English, and informal romanised Tamil is common enough to confuse models trained primarily on formal text. This work presents a system for binary classification of Tamil comments into Abusive and Non-Abusive categories, submitted to the DravidianLangTech@ACL 2026 shared task [12]. MuRIL [5], a BERT-based encoder pre-trained on 17 Indian languages and their transliterated equivalents, is fine-tuned, and it is shown that this Indian-language-specific pre-training provides a meaningful advantage over generic multilingual baselines. The system achieves a macro-averaged F1 of 0.83 on the validation set, compared to 0.79 for XLM-RoBERTa and 0.77 for mBERT under identical training conditions, establishing a strong transformer-based baseline for abusive language detection in code-mixed Tamil.

1 Introduction

Tamil is one of the oldest classical languages still in everyday use, with more than 80 million speakers worldwide. Social media has given Tamil speakers an enormous platform for public expression, but it has also made it easier for abusive and harmful content to spread. Detecting such content automatically is a growing research priority, yet most existing systems are designed for high-resource languages like English and do not transfer well to Tamil.

The challenge goes beyond data scarcity. Tamil is an agglutinative language, where a single word can encode morphological information that would require several words in English. On social media, this complexity is compounded by heavy code-mixing: users blend native Tamil script, romanised Tamil, and English freely within a single comment, with no consistent orthographic convention. Lexicon-based systems and classical machine learning approaches struggle in this setting because they rely on vocabulary stability that code-mixed text simply does not have.

The DravidianLangTech@ACL 2026 shared task on abusive Tamil comment detection [12] provides a structured benchmark for this problem, framing it as binary classification: given a Tamil social media comment, decide whether it is Abusive or Non-Abusive. The approach centers on fine-tuning MuRIL [5], a transformer encoder pre-trained specifically on 17 Indian languages including Tamil, using both native script and transliterated corpora. MuRIL's dual-script coverage makes it uniquely well-suited to the romanised Tamil that is pervasive in online communication.

2 Related Work

Early systems for abusive language detection relied on keyword lists and hand-crafted features combined with traditional classifiers such as Support Vector Machines and Naive Bayes [8]. While effective for formal single-language text, these methods break down when vocabulary is informal or code-mixed.

The introduction of BERT [4] changed the landscape considerably. Pre-trained transformer models produced contextual representations that generalised far better across domains, and their multilingual variants extended these benefits to non-English languages. Ranasinghe and Zampieri [9] demonstrated that fine-tuned multilingual transformers consistently outperform traditional methods on offensive language tasks across multiple languages. Vaswani et al. [14] introduced the Transformer architecture that underpins all of these models, establishing multi-head self-attention as the mechanism for capturing long-range token dependencies. Liu et al. [7] later showed that more robust pre-training strategies yield better downstream

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 331–335
July 3, 2026 ©2026 Association for Computational Linguistics

performance, a finding extended to the multilingual setting by Conneau et al. [3] through XLM-RoBERTa.

For Dravidian languages, the DravidianLangTech and LT-EDI shared tasks have been the primary source of benchmarks. Chakravarthi et al. [2] reported results on offensive language identification in Tamil, Malayalam, and Kannada, where fine-tuned mBERT and XLM-RoBERTa were the dominant approaches. Sai and Sharma [11] explored offensive language identification across Dravidian languages. Tamil-specific work has explored ensemble strategies and model combinations: Anbukkarasi and Varadhaganapathy [1] applied deep learning for hate speech detection in code-mixed Tamil, Subramanian et al. [13] used adapters and cross-domain transfer for offensive language in Tamil YouTube comments, and Krishna et al. [6] addressed class imbalance in Tamil code-mixed abusive comment detection. S et al. [10] applied RoBERTa and XGBoost for abusive Tamil and Malayalam text detection at DravidianLangTech 2025. However, MuRIL has not been systematically evaluated on abusive language detection in Tamil — which is the gap this work addresses.

MuRIL [5] was developed by Google Research and pre-trained on 17 Indian languages using Wikipedia and Common Crawl data in both native and transliterated script. This dual-script design is a key advantage over XLM-RoBERTa and mBERT, neither of which includes transliterated training data. For Tamil social media, where romanised text is widespread, MuRIL's pre-training distribution is a much closer match to the target domain.

3 Dataset and Preprocessing

3.1 Dataset

The dataset was provided by the shared task organisers and consists of Tamil social media comments labelled as Abusive or Non-Abusive. The text reflects real-world Tamil online writing, freely mixing native Tamil script, romanised Tamil, and English within individual comments. This code-mixed, multi-script nature makes automated classification considerably harder than a standard monolingual setting.

The full annotated training set contains 3,652 examples. A stratified 90/10 split is applied to preserve label distribution, yielding 3,286 training examples and 366 validation examples. The test set comprises 912 unlabelled comments for blind evaluation, which is standard practice in shared task settings. The training data exhibits moderate class imbalance: approximately 64% Non-Abusive and 36% Abusive. Table 1 shows the full class distribution. Comment lengths vary from brief single-word utterances to multi-sentence posts, with a mean token length of approximately 18 tokens after Unicode normalisation. The minority Abusive class is the harder of the two to detect reliably, especially when abuse is expressed through sarcasm or indirect phrasing rather than explicit offensive vocabulary.

Label	Count	%
Non-Abusive	2338	64%
Abusive	1314	36%
Total	3652	100%

Table 1: Class distribution for the Tamil abusive comment detection training data.

3.2 Preprocessing

Text samples are tokenised using the MuRIL WordPiece tokeniser, which handles both native Tamil script and romanised equivalents natively. A maximum sequence length of 128 tokens is used, sufficient for the short social media comments in this dataset. Sequences exceeding this limit are truncated, and shorter sequences are padded within each batch. Unicode normalisation is applied to handle encoding inconsistencies common in crawled social media text.

Transliteration, language identification, and stopword removal are deliberately skipped. MuRIL's dual-script pre-training already accounts for the mix of native and romanised Tamil, and additional normalisation risks erasing code-switching cues that carry genuine semantic meaning.

4 Methodology

4.1 System Pipeline

The system follows a straightforward fine-tuning pipeline, illustrated in Figure 1. Each comment is passed to the MuRIL tokeniser, encoded as token IDs with a prepended `[CLS]` token, and fed into the MuRIL transformer encoder. The hidden state at the `[CLS]` position is extracted and passed through a linear classification head that maps it to two logits. These are normalised via softmax, and the class with the highest probability is taken as the prediction.

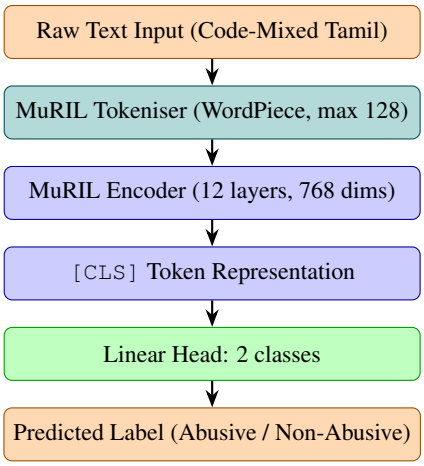


Figure 1: End-to-end system pipeline for abusive comment classification in Tamil.

4.2 Model: MuRIL

The core model is MuRIL (Multilingual Representations for Indian Languages; Khanuja et al. 5), a BERT-based encoder released by Google Research, pre-trained on 17 Indian languages in both native and transliterated script. Its 12-layer transformer architecture produces 768-dimensional contextualised token representations. A single linear classification head is placed over the `[CLS]` token representation, mapping the 768-dimensional vector to two output logits. The entire model is fine-tuned end-to-end.

4.3 Model Formulation

At each transformer layer, MuRIL computes scaled dot-product attention over queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$:

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \quad (1)$$

where d_k is the key vector dimension. This allows the model to capture dependencies between tokens across Tamil script, romanised Tamil, and English simultaneously within a single comment.

The `[CLS]` representation from the final encoder layer is passed through the linear head, and the resulting logits are normalised via softmax to produce class probabilities for $c \in \{\text{Abusive, Non-Abusive}\}$:

$$P(y=c \mid x) = \frac{\exp(z_c)}{\sum_j \exp(z_j)} \quad (2)$$

Training minimises binary cross-entropy loss over all training samples:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\Big[y_i \log P(y{=}1 \mid x_i) + (1-y_i)\log P \quad (3$$

where N is the number of training examples and $y_i \in \{0,1\}$ is the binary ground truth label.

4.4 Training Setup

All models are fine-tuned using the Hugging Face Transformers Trainer API [15] with a learning rate of 2×10^{-5}, batch size of 16, weight decay of 0.01, and 4 training epochs. The AdamW optimiser is used throughout. Macro-averaged F1 on the validation set serves as the primary evaluation metric, since it weights all classes equally and is sensitive to performance on the minority Abusive class. All experiments run on a CUDA-enabled GPU with total training time of approximately 25 minutes.

MuRIL was evaluated at 2, 3, 4, and 5 training epochs on the validation set. Performance improved from 0.79 at 2 epochs to 0.83 at 4 epochs, and stabilised thereafter with no further gain at 5 epochs. The 4-epoch configuration was therefore used for all reported experiments.

5 Results and Discussion

5.1 Official Results

Table 2 reports macro-averaged precision, recall, and F1 on the held-out validation set for MuRIL and two multilingual baselines — mBERT [4] and XLM-R [3] — both fine-tuned under identical settings.

Model	P	R	F1
mBERT	0.78	0.76	0.77
XLM-R	0.80	0.79	0.79
MuRIL (proposed)	**0.84**	**0.83**	**0.83**

Table 2: Macro-averaged results on the validation set (P = Precision, R = Recall, F1 = Macro F1).

MuRIL achieves the best performance across all metrics, with a macro-F1 of 0.83 compared to 0.79 for XLM-R and 0.77 for mBERT. The improvement is most pronounced for the Abusive class, where MuRIL's Indian-language-specific pre-training enables better representation of Tamil morphology and colloquial script variants. The gap between MuRIL and XLM-R is especially telling: XLM-R is a strong multilingual model, but it does not include transliterated training data. For Tamil social

media, that gap in pre-training coverage translates directly into lower recall on abusive content.

Table 3 shows the per-class breakdown for MuRIL, giving a clearer view of where the model performs well and where it still falls short.

Class	Precision	Recall
Non-Abusive	0.87	0.91
Abusive	0.81	0.75

Table 3: Per-class precision and recall for the proposed MuRIL system.

The Non-Abusive class is handled with higher precision and recall, as expected given the class imbalance. A recall of 0.75 on Abusive comments means roughly one in four abusive examples is missed — improving this is the clearest direction for future work.

5.2 Error Analysis

Qualitative examination of the misclassified examples reveals two dominant patterns. The first is implicit abuse: comments that convey abusive intent through sarcasm, euphemism, or cultural reference without any overtly offensive vocabulary. These are consistently misclassified as Non-Abusive by all three models, since the meaning only becomes clear with contextual knowledge the models do not possess.

The second pattern involves dense code-mixing: utterances that switch between Tamil script, romanised Tamil, and English multiple times within a single short comment. When all three layers are simultaneously active, even MuRIL occasionally fails to build a coherent contextual representation.

False negatives (Abusive predicted as Non-Abusive) outnumber false positives across all models, reflecting the majority-class bias introduced by the training imbalance. Future work should explore loss reweighting or focal loss to counteract this tendency, as well as back-translation from related Dravidian languages to augment the Abusive training set.

6 Conclusion

This work presents a system for Tamil abusive comment detection based on fine-tuning MuRIL for binary sequence classification. By leveraging Indian-language-specific pre-training on both native and transliterated scripts, the model achieves a macro-F1 of 0.83 on the validation set, outperforming XLM-R and mBERT by clear margins under identical conditions. The results confirm that pre-training data composition matters: a model exposed to transliterated Tamil is better equipped to handle the romanised code-mixing that defines Tamil social media. Future directions include adversarial training for implicit abuse, ensemble methods combining transformer representations with lexicon-based features, and extension to multi-class offensive language categories.

Limitations

The system is trained and evaluated on the provided shared task dataset alone, and generalisation to other Tamil platforms or time periods cannot be guaranteed. The maximum sequence length of 128 tokens may truncate longer comments, losing relevant context. Computational constraints limited experiments to MuRIL-base. Class imbalance was not addressed through resampling or weighted loss, which likely suppressed recall on the Abusive class.

Acknowledgments

The organizers of the DravidianLangTech@ACL 2026 shared task [12] are thanked for providing the annotated dataset and evaluation infrastructure. Sincere gratitude is expressed to the project guide, S. Sumathi, Department of Information Technology, St. Joseph's College of Engineering, for constant guidance and encouragement throughout this work. The developers of MuRIL and the Hugging Face Transformers library are also thanked for making their tools openly available.

References

[1] S. Anbukkarasi and S. Varadhaganapathy. Deep learning-based hate speech detection in code-mixed Tamil text. *IETE Journal of Research*, 69(11):7893–7898, 2023. URL https://doi.org/10.1080/03772063.2021.1992853.

[2] Bharathi Raja Chakravarthi, Ruba Priyadharshini, Subalalitha Chinnaudayar Navaneethakrishnan, Navaneethan Rajasekaran, Sajeetha Thavareesan, and Dhivya Chinnappa. Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 133–145, 2021. URL https://aclanthology.org/2021.dravidianlangtech-1.17.

[3] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer,

and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, 2020. URL https://aclanthology.org/2020.acl-main.747.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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*, pages 4171–4186, 2019. URL https://aclanthology.org/N19-1423.

[5] 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. MuRIL: Multilingual representations for Indian languages. *arXiv preprint arXiv:2103.10730*, 2021. URL https://arxiv.org/abs/2103.10730.

[6] R. Krishna et al. Abusive comment detection in Tamil code-mixed data by adjusting data distribution. In *Proceedings of the International Conference on Computational Intelligence and Data Engineering (ICCIDE 2024)*, 2024. URL https://doi.org/10.1007/978-981-97-6714-4.

[7] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. URL https://arxiv.org/abs/1907.11692.

[8] Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*, pages 145–153, 2016. URL https://doi.org/10.1145/2872427.2883062.

[9] Tharindu Ranasinghe and Marcos Zampieri. Multilingual offensive language identification with cross-lingual embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5838–5847, 2021. URL https://aclanthology.org/2021.emnlp-main.470.

[10] Nishanth S, Shruthi Rengarajan, S Anantha-sivan, Burugu Rahul, and Sachin Kumar S. ANSR@DravidianLangTech 2025: Detection of abusive Tamil and Malayalam text targeting women on social media using RoBERTa and XGBoost. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 711–715, 2025. URL https://aclanthology.org/2025.dravidianlangtech-1.98.

[11] Siva Sai and Yashvardhan Sharma. Towards offensive language identification for Dravidian languages. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 18–27, 2021. URL https://aclanthology.org/2021.dravidianlangtech-1.3.

[12] Bhuvaneswari Sivagnanam, Kathiravan Pannerselvam, Jananayagan V, Charmathi Rajkumar, Ramesh Kannan R, Ratnavel Rajalakshmi, Shunmuga Priya Muthusamy Chinnan, Saranya Rajiakodi, and Bharathi Raja Chakravarthi. From Comments to Harm: A findings report on abusive Tamil text targeting women on social media. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics, 2026. URL https://aclanthology.org/2026.dravidianlangtech.

[13] S. Subramanian et al. Offensive language detection in Tamil YouTube comments by adapters and cross-domain knowledge transfer. *Computer Speech & Language*, 74:101386, 2022. URL https://doi.org/10.1016/j.csl.2022.101386.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, 2017. URL https://papers.nips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-html.

[15] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, and Jamie Brew. 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, 2020. URL https://aclanthology.org/2020.emnlp-demos.6.

PrimeLine@DravidianLangTech 2026: Hope Speech Detection in Tulu Using XLM-RoBERTa for Coarse and Fine-Grained Classification

Rithikaa V Sanjay Krishnan K Nithya Varshini C N R S. Sumathi
Department of Information Technology, St. Joseph's College of Engineering
https://github.com/Rithikaa7/PrimeLine-Hope_Speech_Detection

Abstract

Hope speech detection in low-resource, code-mixed languages presents a genuine challenge for natural language processing. Tulu, a Dravidian language spoken along the coastal regions of Karnataka and Kerala, is one such language where social media content is deeply code-mixed, blending Tulu, Kannada script, and English within a single comment. Two classification tasks are addressed: a four-class coarse-grained setting (Track 1) and a five-class fine-grained setting (Track 2). XLM-RoBERTa, a cross-lingual transformer pre-trained on more than 100 languages, is fine-tuned on the task-provided datasets using Google Colab with an NVIDIA T4 GPU. The system achieves a Macro F1-score of 0.34 on Track 1 and 0.19 on Track 2 on the official Codabench evaluation, establishing the first transformer-based baseline for hope speech classification in Tulu.

1 Introduction

Hope speech refers to positive, forward-looking online content that encourages, supports, or inspires others. As NLP research has expanded its scope, hope speech detection has grown into a meaningful area of study, offering a constructive counterpart to the widely explored tasks of hate speech and toxicity detection. While considerable effort has been directed at identifying harmful content, the computational identification of positive and hopeful discourse has received comparatively less attention.

Identifying hopeful language in regional and minority languages is particularly valuable, as it gives visibility to communities rarely represented in large-scale language studies. For speakers of languages such as Tulu, computational tools can open up possibilities for social listening and content moderation tailored to their specific needs.

Tulu is spoken by roughly four to five million people in the Tulu Nadu region of coastal Karnataka and Kerala. Despite its significant speaker population, Tulu remains almost entirely absent from NLP research. There is no dedicated pre-trained language model for Tulu and no established NLP pipeline for the language. Its presence on social media is marked by heavy code-mixing, where native Tulu, Kannada script, English, and emoji are blended freely within a single comment, making automated classification considerably harder than equivalent tasks in monolingual settings.

The DravidianLangTech shared task on Hope Speech Detection in Tulu, organized as part of ACL 2026, provides the first structured benchmark for this problem. Track 1 covers four coarse-grained categories: Encouraging Hope, Discouraging Hope, Blended Hope, and Uninvolved. Track 2 covers five fine-grained categories: Inspiring Hope, Hopelessness, Realistic Hope, Optimistic Hope, and Fading Hope. The transition to fine-grained classification makes the task progressively more demanding, as Track 2 categories differ in degree rather than in kind, with genuinely ambiguous boundaries.

The approach taken in this work centers on fine-tuning XLM-RoBERTa [6], a multilingual transformer pre-trained on more than 100 languages. Its cross-lingual representations allow it to handle Tulu's mixed scripts without any language identification or transliteration steps, making it a practical and principled choice for this low-resource, code-mixed setting. The system is trained end-to-end with a linear classification head on top of the encoder, and predictions are generated by taking the argmax of the softmax output.

2 Related Work

Hope speech detection as a formal NLP task was first introduced by Chakravarthi [2] through the HopeEDI dataset, covering English, Tamil, and Malayalam. The dataset and baselines demonstrated that transformer-based models clearly outperform classical approaches even on small mul-

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 336–340
July 3, 2026 ©2026 Association for Computational Linguistics

tilingual corpora, setting the stage for the shared tasks that followed.

Chakravarthi and Muralidaran [4] reported findings from a shared task on hope speech detection across English, Tamil, and Malayalam, confirming that multilingual transformers consistently outperform classical machine learning approaches on code-mixed text.

Several systems from those shared tasks demonstrated the strength of XLM-RoBERTa. Ziehe et al. [16] showed that fine-tuning XLM-RoBERTa achieved strong results across English, Malayalam, and Tamil without any task-specific feature engineering. Mahajan et al. [11] ranked first in both English and Tamil by fine-tuning RoBERTa and XLM-RoBERTa respectively. Balouchzahi et al. [1] built an XLM-RoBERTa ensemble specifically for code-mixed Dravidian text. Huang and Bai [9] combined XLM-RoBERTa with TF-IDF features and ranked first in both English and Malayalam. Singh et al. [13] explored multilingual transformer models pre-trained on Indian languages and achieved second place in both English and Malayalam. These results collectively established XLM-RoBERTa as the dominant baseline for Dravidian hope speech classification.

On the model side, Vaswani et al. [15] introduced the Transformer architecture, which set the foundation for the modern pre-trained language model paradigm. Devlin et al. [7] built on this with BERT, demonstrating the power of masked language modelling for downstream NLP tasks. Liu et al. [10] followed with RoBERTa, showing that more robust pre-training strategies yield better performance. Conneau et al. [6] extended this to the multilingual setting with XLM-RoBERTa, pre-trained on over 100 languages including several Indic scripts.

Dravidian language NLP has received growing attention in recent years. Chakravarthi et al. [5] reported a comprehensive shared task overview on hope speech across five languages. Chakravarthi [3] further presented a multilingual hope speech dataset covering English and Dravidian languages. Sundar et al. [14] proposed a stacked encoder architecture using cross-lingual embeddings and demonstrated improvements over standard fine-tuning baselines.

Research on Tulu specifically is extremely sparse. Shetty [12] examined sentiment analysis in code-mixed Tulu and Tamil text, noting the severe scarcity of annotated data. Ehsan et al. [8] tackled Tulu sentiment classification using BiLSTM models with ELMo embeddings, identifying the lack of Tulu-specific language resources as the primary bottleneck. Neither work addresses hope speech detection in Tulu; the current work is the first to do so.

3 Dataset and Preprocessing

3.1 Dataset

The datasets for both tracks were prepared by the shared task organizers and drawn from social media platforms used by Tulu speakers. The text reflects real-world Tulu online writing, mixing Tulu, Kannada script, English, and emoji freely within individual comments. This code-mixed, multi-script nature is a defining feature of the data and makes classification non-trivial even for human annotators.

Track 1 contains 5,991 training samples across four coarse categories. The distribution is imbalanced, with Uninvolved accounting for 2,490 samples (41.6%) and Discouraging Hope having the fewest at 711 (11.9%). Track 2 has 3,185 training samples across five fine-grained labels, with Inspiring Hope as the majority class at 35.4% and Fading Hope as the minority at 7.4%. Tables 1 and 2 show the full distributions.

Label	Count	%
Encouraging Hope	1895	31.6%
Discouraging Hope	711	11.9%
Blended Hope	895	14.9%
Uninvolved	2490	41.6%
Total	5991	100%

Table 1: Class distribution – Track 1 (Coarse-Grained).

Label	Count	%
Inspiring Hope	1129	35.4%
Hopelessness	937	29.4%
Realistic Hope	503	15.8%
Optimistic Hope	380	11.9%
Fading Hope	236	7.4%
Total	3185	100%

Table 2: Class distribution – Track 2 (Fine-Grained).

3.2 Preprocessing

Preprocessing was kept minimal by design. Text samples were passed directly to the XLM-RoBERTa tokenizer, which handles diverse scripts

through a SentencePiece-based subword vocabulary. Padding and truncation were applied at a maximum length of 128 tokens. Transliteration, language identification, and stopword removal were deliberately skipped, as these steps risk erasing code-switching cues that carry genuine meaning. No normalization of Kannada or Roman script was applied, preserving the natural variation present in user-generated content. This minimal preprocessing strategy is consistent with prior work on code-mixed Dravidian NLP, where aggressive preprocessing has been shown to hurt rather than help performance.

4 Methodology

4.1 System Pipeline

The system follows a standard transformer fine-tuning pipeline. Each raw text sample is first passed to the XLM-RoBERTa tokenizer, which applies SentencePiece subword segmentation and encodes the input as a sequence of token IDs with a special `[CLS]` token prepended. The encoded sequence is fed into the XLM-RoBERTa encoder, which produces contextualized representations through twelve layers of multi-headed self-attention. The hidden state of the `[CLS]` token at the final encoder layer is extracted and passed through a linear classification head. Output logits are converted to class probabilities via softmax, and the predicted label is the class with the highest probability. For Track 1, the output layer has four neurons; for Track 2, five. Figure 1 illustrates the pipeline.

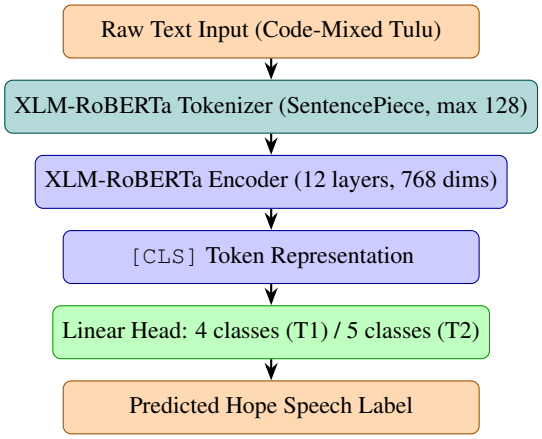


Figure 1: End-to-end system pipeline for hope speech classification in Tulu.

4.2 Model Formulation

The core of the system is `xlm-roberta-base`, a transformer encoder built on the architecture of Vaswani et al. [15] and trained using the robust pre-training strategy of Liu et al. [10] extended to 100+ languages by Conneau et al. [6]. The model contains approximately 270 million parameters and was pre-trained on filtered Common Crawl data spanning 100 languages, including Indic scripts.

Within each transformer layer, the self-attention mechanism computes attention scores over all token pairs. Given query matrix $\mathbf{Q}$, key matrix $\mathbf{K}$, and value matrix $\mathbf{V}$, scaled dot-product attention is:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \quad (1)$$

where d_k is the key vector dimensionality. This allows the model to capture long-range dependencies across mixed scripts and languages simultaneously.

The `[CLS]` representation is fed into a linear head, and logits $z_1, \ldots, z_C$ are normalized via softmax to produce class probabilities:

$$P(y = c \mid x) = \frac{\exp(z_c)}{\sum_{j=1}^{C} \exp(z_j)} \quad (2)$$

The model is trained end-to-end by minimizing categorical cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic} \log P(y = c \mid x_i) \quad (3)$$

where N is the number of training samples, C the number of classes, and $y_{ic} \in \{0, 1\}$ the one-hot ground truth indicator. Gradients are backpropagated through the full encoder, adapting pre-trained representations to the specific vocabulary and style of Tulu social media text.

4.3 Training Setup

Both models were fine-tuned on Google Colab with an NVIDIA T4 GPU for 2 epochs with batch size 8 and AdamW optimizer (learning rate 2×10^{-5}, weight decay 0.01). A linear warmup was applied over the first 6% of training steps, followed by linear decay to zero. The development set served as the validation split throughout, and the best checkpoint by validation loss was selected for inference. Predictions on test sets were obtained by argmax over output logits, converted to label strings, and submitted to Codabench. No ensembling was applied.

5 Results and Discussion

5.1 Official Results

Table 3 reports official Codabench evaluation scores. Accuracy and macro-averaged precision,

recall, and F1 are reported. Macro-averaging is the primary metric as it weights all classes equally, making it sensitive to performance on minority categories regardless of their frequency.

Track	Acc	P	R	F1
Track 1 – Coarse	0.60	0.30	0.41	0.34
Track 2 – Fine	0.24	0.19	0.19	0.19

Table 3: Macro-averaged results on Codabench (P = Precision, R = Recall).

On Track 1, an accuracy of 0.60 is achieved with a Macro F1 of 0.34. The gap between these two figures reflects a model that performs well on the majority Uninvolved class but struggles with less frequent categories. Track 2 is harder, with accuracy at 0.24 and Macro F1 at 0.19, reflecting the semantic closeness of its five fine-grained categories and severe class imbalance.

5.2 Error Analysis

A detailed look at predictions reveals that the accuracy-F1 gap on Track 1 is driven primarily by over-prediction of the Uninvolved class (41.6% of training data). Samples from Blended Hope and Discouraging Hope are prone to being absorbed into this dominant category, as they share surface-level vocabulary with neutral comments. The decision boundary between Encouraging Hope and Blended Hope is especially difficult, since a comment can simultaneously express genuine positivity and realistic acknowledgment of hardship, making clean categorical assignment ambiguous.

Track 2 exhibits a more extreme version of the same problem. The five fine-grained categories are semantically adjacent and differ in degree rather than kind. A comment expressing uncertainty about the future could plausibly belong to Fading Hope, Hopelessness, or Realistic Hope depending on subtle contextual cues difficult to capture with only two epochs of fine-tuning. The scarcity of Fading Hope samples (236 of 3,185) further limits learning for this class.

5.3 Discussion and Future Directions

The results demonstrate that XLM-RoBERTa provides a viable starting point for hope speech detection in Tulu even without Tulu-specific pre-training. Future work should address class imbalance via weighted loss functions or back-translation augmentation from Kannada. Training for more epochs with early stopping based on macro F1 could help balance performance across classes. Larger multilingual models or Dravidian-family-specific pre-trained models could yield better representations for Tulu's lexical patterns. Contrastive learning objectives that explicitly separate semantically similar hope categories could also resolve the fine-grained boundary confusion observed in Track 2.

6 Conclusion

This work presents the first transformer-based system for hope speech detection in code-mixed Tulu, fine-tuned on datasets from the DravidianLangTech@ACL 2026 shared task. XLM-RoBERTa is applied to both tracks, achieving Macro F1 scores of 0.34 and 0.19 on Track 1 and Track 2 respectively. The results establish a meaningful baseline for future work and highlight the dual challenge of class imbalance and fine-grained semantic similarity. Despite modest absolute scores, the cross-lingual transfer capabilities of XLM-RoBERTa prove sufficient to produce non-trivial predictions even for a language with no dedicated NLP resources. It is hoped that this contribution encourages further computational research on Tulu and other low-resource Dravidian languages that remain largely absent from the NLP literature.

Limitations

The primary limitation is the use of only two training epochs, restricting the model's ability to learn reliable decision boundaries for minority classes. The absence of Tulu-specific pre-trained representations means cross-lingual transfer may not fully capture the language's phonological and morphological patterns. Class imbalance was not addressed through resampling or loss-weighting, suppressing performance on rare categories. The system does not perform language identification or script normalization, both avenues for future improvement.

Acknowledgments

The organizers of the DravidianLangTech@ACL 2026 shared task are thanked for providing the datasets and the Codabench evaluation platform. Sincere gratitude is expressed to the project guide, S. Sumathi, Department of Information Technology, St. Joseph's College of Engineering, for constant guidance throughout this work.

References

[1] Fazlourrahman Balouchzahi, Aparna B K, and H. L. Shashirekha. MUCS@LT-EDI-EACL2021: CoHope – hope speech detection for equality, diversity, and inclusion in code-mixed texts. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 180–187, 2021. URL https://aclanthology.org/2021.ltedi-1.25.

[2] Bharathi Raja Chakravarthi. 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, 2020. URL https://aclanthology.org/2020.peoples-1.5.

[3] Bharathi Raja Chakravarthi. Multilingual hope speech detection in English and Dravidian languages. *International Journal of Data Science and Analytics*, 14 (4):389–406, 2022. URL https://doi.org/10.1007/s41060-022-00324-5.

[4] Bharathi Raja Chakravarthi and Vigneshwaran Muralidaran. Findings of the shared task on hope speech detection for equality, diversity, and inclusion. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 61–72, 2021. URL https://aclanthology.org/2021.ltedi-1.8.

[5] Bharathi Raja Chakravarthi et al. Overview of the shared task on hope speech detection for equality, diversity, and inclusion. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 378–388, 2022. URL https://aclanthology.org/2022.ltedi-1.53.

[6] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, 2020. URL https://aclanthology.org/2020.acl-main.747.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186, 2019. URL https://aclanthology.org/N19-1423.

[8] Toqeer Ehsan, Amina Tehseen, Kengatharaiyer Sarveswaran, and Amjad Ali. AlphaBrains@DravidianLangTech: Sentiment analysis of code-mixed Tamil and Tulu using ELMo embeddings. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, pages 152–159, 2023. URL https://aclanthology.org/2023.dravidianlangtech-1.22.

[9] Bo Huang and Yang Bai. TEAM HUB@LT-EDI-EACL2021: Hope speech detection based on pre-trained language model. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 122–127, 2021. URL https://aclanthology.org/2021.ltedi-1.17.

[10] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. URL https://arxiv.org/abs/1907.11692.

[11] Khyati Mahajan, Erfan Al-Hossami, and Samira Shaikh. TeamUNCC@LT-EDI-EACL2021: Hope speech detection using transfer learning with transformers. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 136–142, 2021. URL https://aclanthology.org/2021.ltedi-1.19.

[12] Poorvi Shetty. Poorvi@DravidianLangTech: Sentiment analysis on code-mixed Tulu and Tamil corpus. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, pages 124–132, 2023. URL https://aclanthology.org/2023.dravidianlangtech-1.19.

[13] Pankaj Singh, Prince Kumar, and Pushpak Bhattacharyya. CFILT IIT Bombay@LT-EDI-EACL2021: Hope speech detection using multilingual representation from transformers. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 193–196, 2021. URL https://aclanthology.org/2021.ltedi-1.27.

[14] Arunima Sundar, Akshay Ramakrishnan, Avantika Balaji, and Thenmozhi Durairaj. Hope speech detection for Dravidian languages using cross-lingual embeddings with stacked encoder architecture. *SN Computer Science*, 3:67, 2022. URL https://doi.org/10.1007/s42979-021-00948-3.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, 2017. URL https://papers.nips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845a-Abstract.html.

[16] Stefan Ziehe, Franziska Pannach, and Aravind Krishnan. GCDH@LT-EDI-EACL2021: XLM-RoBERTa for hope speech detection in English, Malayalam, and Tamil. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 132–135, 2021. URL https://aclanthology.org/2021.ltedi-1.18.

RMS@DravidianLangTech 2026: Multimodal Gated Fusion for Hierarchical Tamil Political Meme Classification

Md. Ajwad Hossain
Department Of Electronics & Telecommunication Engineering
Chittagong University Of Engineering & Technology
md.ajwadhossain@gmail.com

Abstract

Internet memes have become a dominant and highly accessible medium for political discourse on social media. However, their multimodal nature—combining culturally specific visual symbols with code-mixed text—presents a significant challenge for automated content analysis, particularly in low-resource languages. In this study, we describe the system submitted by team RMS for the Multi-Level Political Meme Classification shared task at DravidianLangTech @ ACL 2026, focusing exclusively on the Tamil language track. We propose a robust late-fusion multimodal architecture that leverages a pre-trained ResNet-50 network for visual feature extraction and a Transformer-based model (MuRIL) for processing code-mixed Tamil text. The modalities are aligned using bidirectional cross-modal attention and combined using a Gated Multimodal Unit, allowing the model to dynamically weight the importance of visual versus textual cues. Our system ranked 11th on the official leaderboard with a macro-averaged F1-score of 0.7382. Through detailed error analysis, we demonstrate that while our gated fusion approach excels at identifying explicit trolling stances, it struggles with complex target resolution when visual and textual cues contradict.

1 Introduction

The rapid proliferation of political memes on social media has transformed how political narratives are shaped, consumed, and heavily debated. Unlike traditional text-based posts, memes rely on a complex, often sarcastic interplay between visual elements—such as facial expressions, background contexts, and political symbols—and short textual overlays (Kiela et al., 2020). This multimodal combination allows memes to convey rich political stances seamlessly, but it presents a formidable challenge for automated content moderation, particularly in low-resource and highly code-mixed languages like Tamil (Suryawanshi et al., 2020).

To address this gap, the DravidianLangTech @ ACL 2026 workshop introduced the Multi-Level Political Meme Classification shared task (Rajiakodi et al., 2026). The task requires systems to perform a hierarchical classification: first determining the overall stance of the meme (Support vs. Troll/Oppose), and subsequently identifying the specific target of that stance (Individual Person vs. Political Party).

In this study, team RMS presents a robust, multimodal framework designed specifically for the Tamil track of this shared task. While recent advancements in multimodal learning have explored complex cross-attention mechanisms, these models are computationally expensive and highly prone to overfitting on small, noisy datasets. Therefore, we investigate the efficacy of a stable late-fusion approach. Our architecture independently extracts features from the meme's image using a deep residual network (ResNet-50) and the embedded Tamil text using a multilingual Transformer (MuRIL). These representations are subsequently integrated using a Gated Multimodal Fusion layer, which dynamically adjusts the contribution of each modality based on its semantic relevance.

Our empirical results demonstrate that this gated architecture provides a highly competitive baseline for the hierarchical classification task, achieving an overall macro-averaged F1-score of 0.7382 and ranking 11th on the official leaderboard. We make our code publicly available to support future research in low-resource multimodal content analysis. [1]

2 Related Work

Multimodal Meme Analysis: The automated detection of trolls, hate speech, and sentiment in internet memes has gained significant traction. Early approaches primarily relied on unimodal text analysis,

[1] Code is available on GitHub.

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 341–347
July 3, 2026 ©2026 Association for Computational Linguistics

applying standard NLP techniques to extracted Optical Character Recognition (OCR) text. However, Kiela et al. (2020) demonstrated that memes are often "benign confounders," where the text and image are harmless independently but toxic when combined, necessitating true multimodal architectures. In the context of Dravidian languages, Suryawanshi et al. (2020) pioneered the creation of Tamil meme datasets, highlighting the unique challenges of parsing code-mixed Tanglish text alongside culturally specific visual humor.

Code-Mixed Text and Visual Encoding: Processing Tanglish text requires models capable of handling transliteration and code-switching without relying on strict syntactic rules. Khanuja et al. (2021) introduced MuRIL, a BERT-based representation explicitly pre-trained on Indian languages, which has become the standard for such tasks. For visual encoding, deep convolutional networks like ResNet (He et al., 2016) remain the backbone for extracting high-level features from noisy meme templates.

Multimodal Fusion Strategies: The crux of meme classification lies in how the modalities are fused. Simple concatenation (early fusion) often suffers from modality dominance, where a strong visual signal overshadows textual nuances. While recent transformer-based cross-attention models align modalities effectively, they require massive datasets to train. As an alternative, Arevalo et al. (2017) proposed Gated Multimodal Units (GMU), which learn to selectively weight visual and textual features. Our work adapts this gating philosophy, utilizing it to stabilize training on a highly imbalanced, low-resource political meme dataset.

3 Task and Dataset Description

The Multi-Level Political Meme Classification shared task (Rajiakodi et al., 2026) provides a dataset of political memes in Dravidian languages. Our team participated exclusively in the Tamil language track. The task is structured hierarchically into two distinct levels of classification:

Level 1 (Stance Detection): This primary task requires classifying the overall sentiment or stance of the multimodal meme. The system must predict whether the meme expresses a 'Support' stance or a 'Troll/Oppose' stance towards its subject.

Level 2 (Target Identification): For memes identified in Level 1, this secondary task requires fine-grained classification to determine the specific target of the stance. The target classes include broad categories such as 'Individual Person' and 'Political Party'.

The dataset presents unique challenges, as the memes feature culturally specific Tamil political imagery combined with code-mixed Tanglish (Tamil and English) text, often utilizing heavy sarcasm where the visual and textual modalities may present contradictory sentiments. A critical challenge of this dataset is the severe class imbalance present in the training distribution. In Level 1, the data is heavily skewed toward the *Troll/Oppose* class (691 instances) compared to the *Support/Praise* class (112 instances). This imbalance cascades into Level 2, where *Troll/Oppose Against Person* (547 instances) heavily dominates minority classes such as *Support for party* (24 instances), requiring the model to learn robust representations from very few positive examples.

4 Methodology

Our proposed system employs a sophisticated late-fusion multimodal architecture to hierarchically classify political memes. To address the semantic gap between visual and textual modalities, we extract deep representations independently and align them using a Bidirectional Cross-Modal Attention mechanism. The aligned features are then dynamically weighted using a Gated Multimodal Unit before being passed to hierarchical classification heads.

4.1 Text Extraction and Preprocessing

Since internet memes embed text directly within the image payload, robust Optical Character Recognition (OCR) is a prerequisite for multimodal learning. To extract the code-mixed Tamil and English text from the noisy meme backgrounds, we utilized the `ocr_tamil` library, which is specifically optimized for Dravidian scripts (Baek et al., 2019; Bautista and Atienza, 2022; D, 2024).

The raw OCR output frequently contains internet artifacts and structural noise that degrade the performance of transformer-based language models. To mitigate this, we implemented a comprehensive text-cleaning pipeline. Our preprocessing sequence systematically removes URLs, email addresses, and non-alphanumeric special characters. Additionally, we normalized excessive whitespaces, converted all text to lowercase, and truncated consecutive duplicate characters—a common fea-

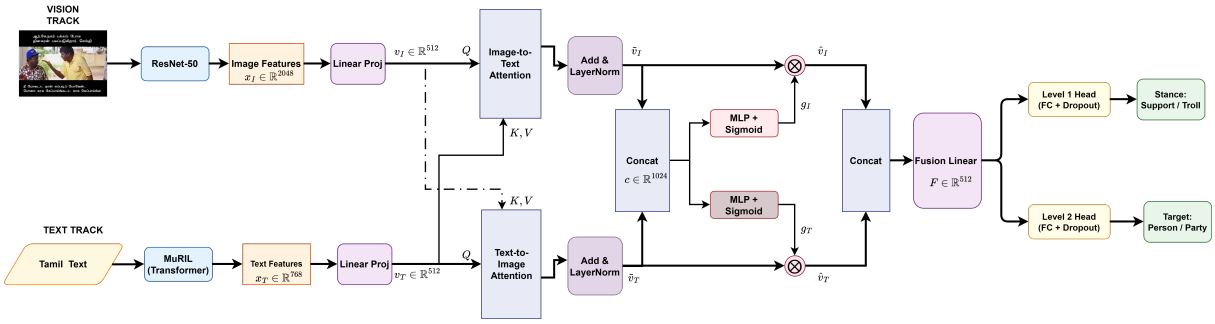


Figure 1: The proposed Gated Cross-Modal architecture utilized by Team RMS. The model aligns visual and textual features via cross-attention and dynamically fuses them using modality-specific gating mechanisms.

ture in memes used for comedic emphasis (e.g., "loool")—to a maximum of two repetitions. In instances where the OCR engine failed to detect any text, a fallback generic token (`meme_image`) was supplied to the textual encoder to ensure dimensional stability during batch processing.

4.2 Modality Encoders

Visual Encoder: To capture the visual semantics of the memes, including facial expressions and political symbols, we utilize a pre-trained ResNet-50 network. We remove the final classification head and extract the flattened output from the global average pooling layer. For a given input image I, the raw visual feature vector is:

$$x_I = \text{ResNet50}(I) \quad \text{where} \quad x_I \in \mathbb{R}^{2048} \tag{1}$$

Textual Encoder: The textual components of the memes are highly code-mixed Tamil (Tanglish) and English. We employ MuRIL, a Transformer model specifically pre-trained on Indian text corpora. For a given caption T, we tokenize the sequence and extract the pooled `[CLS]` token embedding:

$$x_T = \text{MuRIL}(T) \quad \text{where} \quad x_T \in \mathbb{R}^{768} \tag{2}$$

4.3 Bidirectional Cross-Modal Attention

Before fusing the modalities, we align them into a shared semantic space to allow the visual and textual features to inform one another. The raw modality vectors are linearly projected into a shared dimension $d = 512$:

$$v_I = W_{pI} x_I + b_{pI}, \quad v_T = W_{pT} x_T + b_{pT} \tag{3}$$

We employ residual connections and Layer Normalization (LN). Let $\text{MHA}(Q, K, V)$ denote the Multi-Head Attention operation. The cross-attended features are computed as:

$$\tilde{v}_I = \text{LN}(v_I + \text{MHA}(v_I, v_T, v_T)) \tag{4}$$

$$\tilde{v}_T = \text{LN}(v_T + \text{MHA}(v_T, v_I, v_I)) \tag{5}$$

Here, $\tilde{v}_I$ and $\tilde{v}_T \in \mathbb{R}^{512}$ represent the cross-attended image and text features.

4.4 Gated Multimodal Fusion

To dynamically control the contribution of each modality, we introduce a Gated Fusion module. We concatenate the attended features to form a joint representation $c \in \mathbb{R}^{1024}$:

$$c = [\tilde{v}_I; \tilde{v}_T] \tag{6}$$

This representation is passed through two parallel Multi-Layer Perceptrons (MLPs) to generate modality-specific scalar gates. Each MLP uses a Sigmoid activation (σ) to constrain the output to $(0, 1)$:

$$g_I = \sigma(W_{gI2}(\text{ReLU}(W_{gI1} c + b_{gI1})) + b_{gI2}) \tag{7}$$

$$g_T = \sigma(W_{gT2}(\text{ReLU}(W_{gT1} c + b_{gT1})) + b_{gT2}) \tag{8}$$

The attended features are then scaled by their respective learned gates:

$$\hat{v}_I = \tilde{v}_I \cdot g_I, \quad \hat{v}_T = \tilde{v}_T \cdot g_T \tag{9}$$

Finally, the gated features are concatenated and linearly projected back to $d = 512$ to form the unified multimodal embedding F:

$$F = W_F[\hat{v}_I; \hat{v}_T] + b_F \tag{10}$$

4.5 Hierarchical Classification

The unified representation F is fed into two parallel classification heads. Each head consists of a fully connected layer, a ReLU activation, Dropout ($p = 0.3$), and a final linear projection.

For the Level 1 task (Stance Detection), the probability distribution over the classes is computed as:

$$h_{L1} = \text{Dropout}(\text{ReLU}(W_{1a}F)) \quad (11)$$

$$\hat{p}_{L1} = \text{softmax}(W_{1b}h_{L1} + b_{1b}) \quad (12)$$

A similar formulation is used to yield $\hat{p}_{L2}$ for the Level 2 task. The network is optimized jointly using an unweighted sum of categorical cross-entropy losses:

$$\mathcal{L}_{total} = \mathcal{L}_{CE}(\hat{y}_{L1}, y_{L1}) + \mathcal{L}_{CE}(\hat{y}_{L2}, y_{L2}) \quad (13)$$

5 Results and Error Analysis

5.1 Experimental Results

Implementation Details: All architectural variants were trained for 15 epochs using a learning rate of 4×10^{-5}. To ensure strict reproducibility across all internal validations and final test-set evaluations, the random seed was universally fixed to 42.

Table 1 presents the official test set performance of our Gated Multimodal Fusion system. The model achieved an F1-score of 0.8608 on the Level 1 stance detection task, demonstrating a strong capability to distinguish between standard Support and Troll memes. Performance on the more granular Level 2 target identification task yielded an F1-score of 0.6156, reflecting the inherent difficulty of resolving specific political targets in highly code-mixed, low-resource data. Overall, the system achieved a macro-averaged F1 of 0.7382, securing the 11th position among the participating systems.

While we conducted preliminary internal evaluations and hyperparameter tuning using a standard train-validation split, we found that unimodal and un-gated architectures were prone to rapid overfitting. As detailed in our comprehensive ablation study (see Appendix A), the decision to employ a gated cross-modal approach provided the necessary regularization to maintain stable validation trajectories. Furthermore, visual evidence of model capacity from these ablation curves strictly motivated our final submission strategy: retraining the selected architecture on the entirety of the provided dataset to maximize exposure to minority class features before generating predictions for the official test set.

Task	Precision	Recall	F1-Score
L1 (Stance)	0.8554	0.8756	0.8608
L2 (Target)	0.6151	0.6318	0.6156
Avg F1	**0.7382 (Rank: 11th)**		

Table 1: Official test set results for the RMS model.

5.2 Error Analysis

Class Imbalance: The most prominent source of error stems from the severe skew in the training data. The model demonstrated excellent recall for the dominant *Troll/Oppose* class (which constitutes roughly 86% of the training data) but struggled significantly with the minority *Support* classes. When cross-modal signals were ambiguous, the model heavily favored the majority prior, frequently defaulting to a Troll prediction.

Target Confusion (Person vs. Party): In Level 2, the model frequently misclassified 'Troll against Party' as 'Troll against Individual Person'. Memes criticizing a political party almost always feature a recognizable photograph of that party's leader. Because our visual encoder is adept at facial feature extraction, the presence of a face strongly triggered the visual pathway to output representations aligned with the 'Individual Person' class. The textual encoder occasionally failed to extract the specific party-level nuances from the Tanglish text to override this strong visual signal.

6 Conclusion

In this study, we described the system submitted by team RMS for the Multi-Level Political Meme Classification shared task. We designed a late-fusion multimodal architecture that aligns deep visual and textual representations using Bidirectional Cross-Modal Attention, followed by a Gated Multimodal Unit. Our approach achieved competitive results, ranking 11th on the official Tamil track leaderboard. Our analysis revealed a distinct bias in the visual pathway, where the presence of a politician's face often caused the model to misclassify party-directed trolling. Future work should focus on explicit entity-grounding techniques that allow textual cues to override dominating visual signals.

Limitations

While our gated fusion architecture demonstrates competitive performance, several limitations exist. First, the model is trained and evaluated exclusively on the provided Tamil dataset; its generalizability to the Malayalam track or other cross-

lingual zero-shot scenarios remains untested. Second, as highlighted in our error analysis, the visual encoder exhibits a strong prior toward facial features, which can overpower textual cues in complex code-mixed scenarios. Addressing this requires more sophisticated visual-textual grounding mechanisms. Finally, the severe class imbalance restricts the model's reliability on minority classes (e.g., 'Support for party'), indicating that future work must explore advanced re-weighting strategies or synthetic data generation for low-resource Tanglish text.

Finally, while our gated late-fusion approach establishes a computationally efficient baseline, future iterations would significantly benefit from exploring recent advancements in Multimodal Large Language Models (MLLMs). Foundation models such as CLIP (Radford et al., 2021) and LLaVA (Liu et al., 2023) offer promising avenues for transfer learning, which could improve generalizability across low-resource Dravidian languages. Leveraging these large-scale vision-language models could potentially resolve the modality dominance issues we observed by providing deeper semantic grounding when visual and textual cues contradict.

Ethical considerations

The automated classification of political memes carries significant ethical implications. While systems like ours are designed to assist in content moderation and prevent the spread of malicious trolling, they must be deployed with caution to avoid silencing legitimate political dissent or satire. Furthermore, we acknowledge that the dataset contains offensive content; all data was processed strictly for research purposes following the shared task guidelines.

Finally, to prevent misuse, the approach should be viewed as a human-in-the-loop assistive tool rather than an autonomous moderation system.

References

John Arevalo, Thamar Solorio, Manuel Montes y Gómez, and Fabio A. González. 2017. Gated multimodal units for information fusion. *Preprint*, arXiv:1702.01992.

Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. 2019. Character region awareness for text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9365–9374.

Darwin Bautista and Rowel Atienza. 2022. Scene text recognition with permuted autoregressive sequence models. In *European Conference on Computer Vision*, pages 178–196, Cham. Springer Nature Switzerland.

Gnana Prasath D. 2024. Tamil ocr.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

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.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: detecting hate speech in multimodal memes. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Saranya Rajiakodi, Shunmuga Priya Muthusamy Chinnan, Premjith B, Subalalitha CN, Rahul Ponnusamy, Anshid K A, Bhuvaneshwari Sivagnanam, Jananayagan V, Bharathi Raja Chakravarthi, Ragavan N, and Santhini P. 2026. Overview of the shared task on multilevel political meme classification in tamil and malayalam. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Shardul Suryawanshi, Bharathi Raja Chakravarthi, Mihael Arcan, and Paul Buitelaar. 2020. Multimodal meme dataset (MultiOFF) for identifying offensive content in image and text. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 32–41, Marseille, France. European Language Resources Association (ELRA).

A Ablation Study and Training Dynamics

To isolate the contributions of the visual and textual modalities, we conducted a comprehensive ablation study utilizing a 25% internal validation split. As shown in Table 2, we evaluated four architectural variations, recording both accuracy and macro F1-scores at the final epoch.

Architecture	Level 1		Level 2	
	Acc	F1	Acc	F1
Image-Only	0.9104	0.8997	0.6965	0.6511
Attention-Only	0.9104	0.8973	0.6915	0.6261
Gated Fusion	**0.9204**	**0.9109**	**0.6915**	**0.6470**
Cross-Modal (Ours)	0.9104	0.8973	0.6816	0.6421

Table 2: Final epoch validation metrics for architectural variants.

Due to the limited size of the validation split and the severe visual bias inherent in the meme dataset, performance metrics exhibited high variance. As illustrated in Figure 2, plotting the training versus validation accuracy on a fixed scale reveals severe overfitting trajectories across all models. Training accuracy rapidly converges to 1.0 by Epoch 5, while validation accuracy plateaus early.

This pronounced train-validation gap strongly indicated that our models possessed excess capacity for the small training set. Consequently, while the Gated Cross-Modal architecture was selected for its theoretical capacity to resolve complex visual-textual contradictions, this visual evidence strictly necessitated our final competition strategy: retraining the selected model on the entirety of the combined dataset before inference to maximize exposure to the highly skewed minority classes.

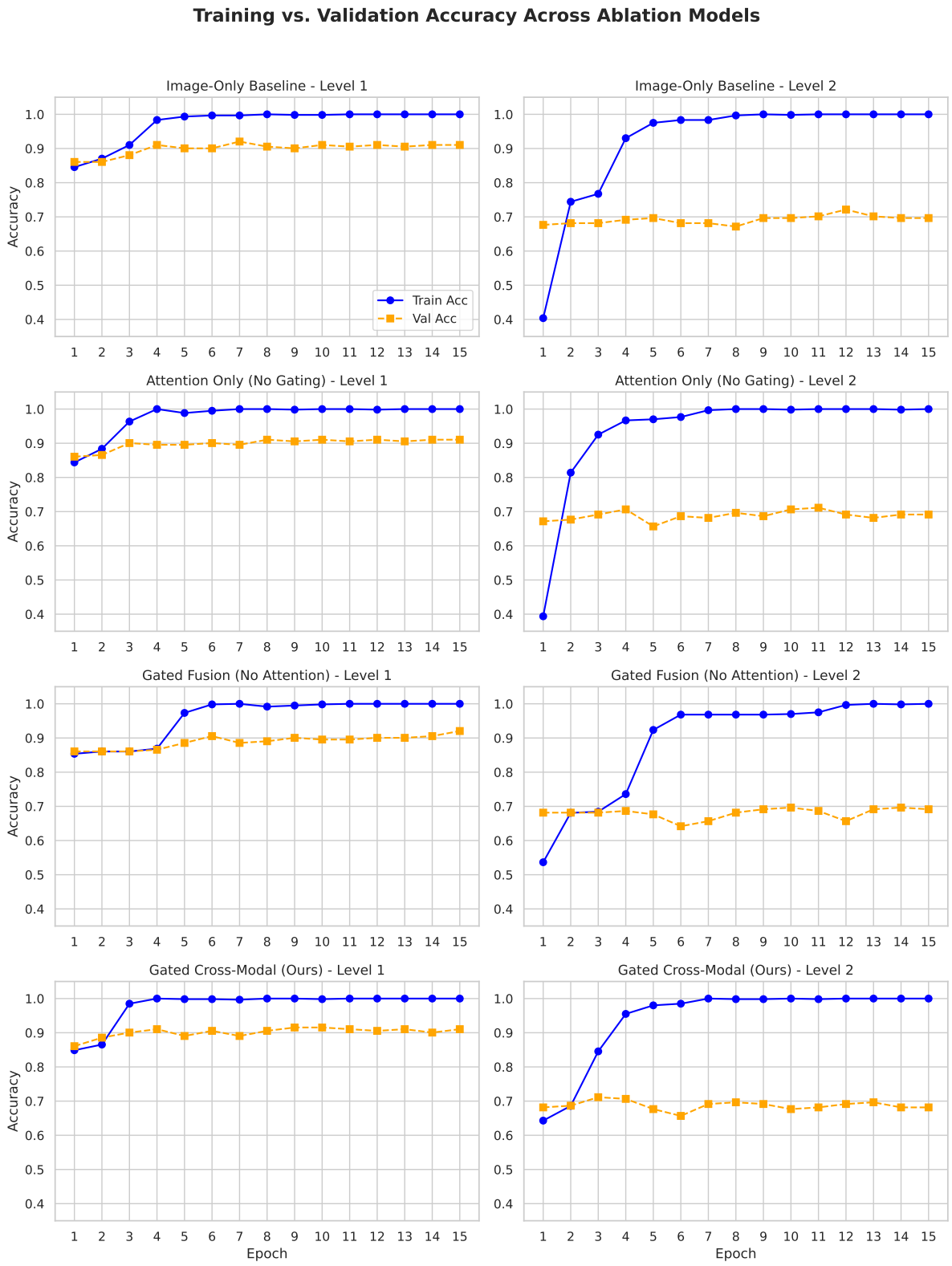


Figure 2: Training versus Validation Accuracy across 15 epochs for all four architectural variants (Level 1 left, Level 2 right). The curves demonstrate the aggressive overfitting dynamic, motivating our final submission strategy of retraining the selected architecture on the available training data.

Semantica@DravidianLangTech 2026: Vision-Language Models for Hierarchical Political Meme Classification in Tamil and Malayalam

Junain Uddin, Rahul Datta, Taha Ibne Abdullah, Hasan Murad
Department of Computer Science and Engineering,
Chittagong University of Engineering and Technology, Bangladesh
{u2104040, u2104048, u2104052}@student.cuet.ac.bd, hasanmurad@cuet.ac.bd
*

Abstract

Political memes are widely used to express opinions, sarcasm, and ideological narratives on social media platforms. However, detecting political trolling in low-resource languages such as Tamil and Malayalam remains challenging due to limited datasets and tools. To address this problem, DravidianLangTech@ACL 2026 organized a shared task on hierarchical political meme classification.

This work explores text-only models, classical multimodal fusion, and Vision-Language Models (VLMs) for Tamil and Malayalam political meme classification. Our experiments include IndicBERTv2, XLM-RoBERTa, EfficientNet-based multimodal fusion, and Qwen-VL models. Among the submitted systems, Qwen2.5-VL-7B-Instruct with 4-bit QLoRA fine-tuning achieved competitive performance, ranking 3rd in the Malayalam track and 4th in the Tamil track based on weighted-F1 score. Additional post-evaluation experiments with Qwen3-VL-8B further improved macro-F1 performance, highlighting the effectiveness of VLMs for low-resource multilingual political meme classification.

1 Introduction

Memes are widely used to express humor, sarcasm, and opinions through combined visual and textual content (Hegde et al., 2021), and have become a common form of communication on social media platforms. While many memes are created for entertainment, political memes are increasingly used to target individuals, parties, or ideologies through trolling and subtle forms of online harassment (Das et al., 2022). Such content can amplify polarized narratives and influence societal perceptions and democratic discussions, making NLP and AI techniques important for analyzing multimodal political content at scale.

Most existing studies mainly focus on high-resource languages such as English (Liu et al., 2025; Huertas-Tato et al., 2024), while comparatively less attention has been given to low-resource languages like Tamil, Malayalam, and Bangla. In addition, prior works primarily focus on hateful or harmful meme detection (Hossain et al., 2024; Pramanick et al., 2021), with limited emphasis on fine-grained political stance classification in multilingual settings.

To address this challenge, DravidianLangTech@ACL 2026 organized a shared task on "Multi-Level Political Meme Classification" (Rajiakodi et al., 2026) for Tamil and Malayalam political memes. The task aims to encourage the development of culturally aware multimodal systems capable of identifying political stance and target information from meme discourse in low-resource Dravidian languages. The main contributions of this work are as follows:

- We compare text-only, classical multimodal fusion, and Vision-Language Models for hierarchical political meme classification in Tamil and Malayalam.
- We analyze the impact of severe class imbalance on multimodal political meme classification across Level-1 and Level-2 tasks.
- We demonstrate the effectiveness of Vision-Language Models for low-resource multilingual political meme understanding.

2 Related Work

Multimodal meme analysis gained prominence with the Hateful Memes Challenge (Kiela et al., 2020), which highlighted the importance of jointly modeling visual and textual signals. Early cross-modal transformers such as ViLBERT (Lu et al., 2019) introduced attention-based alignment mech-

*https://github.com/Junainn/Shared-Task-2026

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 348–353
July 3, 2026 ©2026 Association for Computational Linguistics

Language	Train	Test
Tamil	803	201
Malayalam	500	100
Total	1303	301

Table 1: Language-wise train/test distribution.

anisms for vision-language reasoning, forming the foundation of modern multimodal architectures.

In Dravidian languages, prior shared-task systems explored troll and misogyny meme detection using multimodal fusion techniques (Hegde et al., 2021; Das et al., 2022; Hossan et al., 2025). However, fine-grained political meme classification with stance and target prediction remains relatively underexplored.

Large vision-language models such as Qwen-VL (Bai et al., 2023) extend transformer-based multimodal reasoning to diverse downstream tasks. Parameter-efficient methods like LoRA (Hu et al., 2022), along with memory-efficient optimization techniques including 8-bit quantization (Dettmers et al., 2022) and AdamW optimization (Loshchilov and Hutter, 2019), enable practical adaptation of large multimodal models in low-resource settings.

Despite strong general-domain performance, large multimodal models remain underexplored for political meme classification involving cultural nuance, multilingual text, and severe class imbalance in low-resource languages.

3 Data Description

The dataset consists of political memes in two Dravidian languages: Tamil and Malayalam. Each sample contains a meme image, associated textual content, and hierarchical annotations. The task is formulated as a two-level classification problem: Level-1 represents coarse-grained stance (Troll/Oppose vs. Support), while Level-2 provides fine-grained target-specific labels.

Dataset Split: The dataset is divided into training and test sets for both languages, with a validation split created from the training data. Table 1 summarizes the overall distribution.

Level-1 Distribution: Tamil contains 691 Troll/Oppose and 112 Support/Praise samples, while Malayalam contains 477 Troll/Oppose and 23 Support/Praise samples, indicating severe imbalance in both datasets.

Level-2 Distribution: Tamil contains four fine-grained classes, while Malayalam contains five classes including an additional *Intersection* category. Tables 2 and 3 show the Level-2 training distribution.

Tamil Level-2 Class	Train
Troll/Oppose Against Person	547
Troll/Oppose Against Party	146
Support for person	86
Support for party	24

Table 2: Tamil Level-2 class distribution.

Malayalam Level-2 Class	Train
Against individual person	315
Against party	110
Intersection	53
Support for individual person	12
Support for party	10

Table 3: Malayalam Level-2 class distribution.

4 Methodology

We explored three modeling paradigms: (i) text-only models, (ii) classical multimodal fusion, and (iii) large Vision-Language Models (VLMs). All approaches were evaluated under identical train-test splits for both Level-1 and Level-2 classification tasks.

4.1 System Overview

The objective of exploring multiple paradigms is to systematically analyze the contribution of textual information, visual features, and large-scale cross-modal pretraining in political meme classification. Figure 1 shows the overall experimental pipeline used in this study.

4.2 Text-Only Approach

The text-only baseline evaluates classification using OCR-extracted text without visual features, employing a hierarchical two-stage strategy.

OCR Extraction and Hierarchical Classification: Text was extracted from meme images using Qwen2-VL-2B-Instruct with the prompt *"Extract all text from this image. Return only the text."* Outputs were cleaned using regex to remove conversational artifacts and cached to avoid recomputation. For hierarchical classification, we first predicted Level-2 (fine-grained) labels from OCR text, then prepended the prediction to the original text as `[Level2: <prediction>] <ocr_text>` for Level-1 classification. This allows coarse-level predictions to benefit from fine-grained contextual signals.

Models and Training: We fine-tuned two transformer encoders: ai4bharat/IndicBERTv2-MLM-

Figure 1: Overall experimental pipeline showing text-only, multimodal fusion, and VLM-based approaches.

only and xlm-roberta-base, with separate models for Level-1 and Level-2. Training used 15 epochs on an 85/15 stratified split with batch size 16, learning rate 1×10^{-5}, weight decay 0.01, warmup ratio 0.1, and max sequence length 128. Class imbalance was handled via weighted cross-entropy loss with FP16 mixed precision. Best checkpoints were selected by validation macro F1-score, which was also the primary evaluation metric alongside accuracy and per-class metrics.

4.3 Classical Multimodal Fusion

The classical multimodal baseline models visual and textual information jointly using an early-fusion architecture without relying on OCR.

Visual and Text Encoders: Images were encoded using pretrained EfficientNet-B3 (Tan and Le, 2019) (ImageNet pretrained). The final classification layer was removed, and the 1536-dimensional pooled feature was projected to a 512-dimensional representation via a linear layer with ReLU. For text, *bert-base-multilingual-cased* (mBERT) (Devlin et al., 2019) was used with bilingual political vocabulary prompts containing commonly used political terms, party references, and stance-related keywords in both English and Tamil/Malayalam. These prompts were intended to encourage politically relevant multilingual representations within mBERT. The 768-dimensional [CLS] representation served as the textual feature.

Fusion and Training: The 512-dimensional visual and 768-dimensional textual embeddings were concatenated into a 1280-dimensional joint representation and passed through a small MLP (Dropout $\rightarrow$ Linear $\rightarrow$ ReLU $\rightarrow$ Dropout $\rightarrow$ Linear) for classification. Training followed a three-phase schedule: phases 1–2 (LR=2×10^{-4}, 5×10^{-5}) updated all layers, while phase 3 (LR=1×10^{-5}) froze mBERT and continued training the visual and fusion layers. Optimization used AdamW with class-weighted cross-entropy loss and a OneCycle scheduler, with strong image augmentation and class-balanced oversampling to address data imbalance.

4.4 Vision-Language Models

We fine-tuned instruction-tuned Vision-Language Models from the Qwen-VL family (Bai et al., 2023). While Qwen3-VL-8B achieved stronger post-evaluation performance, Qwen2.5-VL-7B was used for the official shared-task submission, as the Qwen3 experiments were conducted later during additional evaluation and analysis.

Architecture and Optimization: Qwen-VL processes images and text jointly within a unified transformer, enabling cross-modal reasoning without explicit feature fusion. The models can directly interpret Tamil and Malayalam text within images through multimodal pretraining, eliminating the need for OCR preprocessing. Models were loaded in 4-bit quantized format (Dettmers et al., 2022) with gradient checkpointing to enable efficient fine-tuning on limited GPU resources. Table 4 shows the training parameters used in this study.

Parameter-Efficient Fine-Tuning: We adopted Low-Rank Adaptation (LoRA) (Hu et al., 2022), inserting adapters into vision layers, language layers, attention modules, and MLP blocks.

Data Balancing and Training: To address severe class imbalance (23:1 for Tamil and 32:1 for Malayalam), minority classes were oversampled to approximately 200 samples per class. Training followed a supervised instruction-tuning setup, where each meme image was paired with a natural language classification prompt. The prompts explicitly defined each class label (e.g., *troll_person*, *troll_party*, and *support_person*) and instructed the model to generate only the final category name. For Malayalam, additional prompt clarification was added for the *intersection* class to better distinguish memes targeting both a political individual and their party, particularly in sarcastic or ambiguous cases. Tamil experiments converged in one epoch, while Malayalam required two epochs with a reduced learning rate in the second stage to refine

Parameter	Tamil	Malayalam
Quantization	4-bit	4-bit
LoRA rank (r)	16	16
LoRA alpha (α)	16	16
Batch Size	2	2
Learning Rate	2e-5	2e-5 (epoch 1) 5e-6 (epoch 2)
Epochs	1	2
Optimizer	AdamW (8-bit)	AdamW (8-bit)
LR Schedule	Cosine	Cosine

Table 4: VLM fine-tuning configuration

Model	Accuracy	Macro-F1	Weighted-F1
Tamil (201 samples)			
IndicBERTv2 (Text-only)	38.31	28.68	43.00
XLM-RoBERTa (Text-only)	44.28	33.87	47.00
mBERT + EfficientNet (Multimodal)	63.18	34.59	58.39
Qwen2.5-VL-7B	66.67	45.85	67.85
Qwen3-VL-8B	**70.15**	**49.70**	**69.71**
Malayalam (100 samples)			
IndicBERTv2 (Text-only)	48.00	27.12	42.00
XLM-RoBERTa (Text-only)	42.00	30.54	41.00
mBERT + EfficientNet (Multimodal)	36.00	22.16	34.11
Qwen2.5-VL-7B	49.00	35.94	49.78
Qwen3-VL-8B	**57.00**	**36.40**	**58.29**

Table 5: Level-2 classification performance (%) across text-only, multimodal, and VLM approaches.

class boundaries. Inference used greedy decoding for deterministic predictions.

5 Results and Analysis

5.1 Level-2 Classification Results

Level-2 classification focuses on fine-grained stance categories in political memes. Table 5 compares text-only models, classical multimodal fusion, and vision-language models.

Across both languages, models that incorporate visual information perform better than text-only baselines. The best results are obtained by Qwen3-VL-8B, which achieves a macro-F1 of 49.70% for Tamil and 36.40% for Malayalam. This improvement suggests that jointly modeling visual and textual cues is important for capturing the nuanced meaning of political memes.

5.2 Level-1 Classification Results

Level-1 classification focuses on binary stance detection (Support/Praise vs. Troll/Oppose). Table 6 reports results across Tamil and Malayalam.

For Tamil, models that incorporate visual information perform better than text-only baselines, with Qwen3-VL-8B achieving the best macro-F1 score of 82.12%.

Model	Accuracy	Macro-F1	Weighted-F1
Tamil (201 samples)			
IndicBERTv2 (Text-only)	67.16	51.18	72.00
XLM-RoBERTa (Text-only)	78.11	56.90	79.00
mBERT + EfficientNet (Multimodal)	81.59	67.63	83.39
Qwen2.5-VL-7B	87.06	75.84	88.05
Qwen3-VL-8B	**90.55**	**82.12**	**91.22**
Malayalam (100 samples)			
IndicBERTv2 (Text-only)	96.00	48.98	94.00
XLM-RoBERTa (Text-only)	96.00	48.98	94.00
mBERT + EfficientNet (Multimodal)	92.00	31.94	92.00
Qwen2.5-VL-7B	88.00	31.54	90.84
Qwen3-VL-8B	**85.00**	**30.80**	**88.70**

Table 6: Level-1 binary classification performance (%) across text-only, multimodal, and VLM approaches.

Malayalam results are strongly affected by severe class imbalance, with only four Support/Praise instances compared to ninety-six Troll/Oppose samples in the test set. As a result, text-only models achieve high accuracy (96%) by predicting the majority class, while macro-F1 reveals poor minority-class performance.

Overall, Level-1 classification is substantially easier than the fine-grained Level-2 task, leading to higher scores across all models.

6 Conclusion

This work investigated hierarchical political meme classification for Tamil and Malayalam using text-only models, classical multimodal fusion, and Vision-Language Models (VLMs). The best-performing results were achieved by Qwen3-VL-8B, which obtained the highest macro-F1 and weighted-F1 scores across both Tamil and Malayalam datasets in the Level-1 and Level-2 classification tasks.

Our analysis also highlights severe class imbalance in the Malayalam dataset, where accuracy can be misleading despite poor minority-class performance. This emphasizes the importance of macro-averaged metrics for reliable evaluation.

Overall, jointly modeling visual and textual information improves political meme classification in low-resource languages.

Limitations

The provided dataset was relatively small, especially for the malayalam dataset. Both datasets were extremely imbalanced which impacted the predictions for minority classes. The performance was constrained by limited access to GPU resources, which restricted the use of more computationally intensive and powerful models.

Ethical Statement

All sources referenced in this study have been appropriately cited, and care was taken to ensure the originality and integrity of the work. Through this research, we aim to contribute to improved detection of harmful political memes, with the broader objective of helping mitigate online trolling and bullying. The study was conducted in accordance with ethical research standards, and the dataset was used strictly for academic purposes without promoting or endorsing any political ideology.

Acknowledgement

We express our gratitude to the DravidianLangTech 2026 shared task organizers for facilitating this evaluation campaign.

References

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Mithun Das, Somnath Banerjee, and Animesh Mukherjee. 2022. hate-alert@dravidianlangtech-acl2022: Ensembling multi-modalities for tamil trollmeme classification. *Preprint*, arXiv:2204.12587.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. 8-bit optimizers via block-wise quantization. *arXiv preprint arXiv:2110.02861*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Siddhanth U Hegde, Adeep Hande, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. 2021. Uvce-iiitt@dravidianlangtech-eacl2021: Tamil troll meme classification: You need to pay more attention. *Preprint*, arXiv:2104.09081.

Eftekhar Hossain, Omar Sharif, Mohammed Moshiul Hoque, and Sarah M. Preum. 2024. Deciphering hate: Identifying hateful memes and their targets. *Preprint*, arXiv:2403.10829.

Md. Refaj Hossan, Nazmus Sakib, Md. Alam Miah, Jawad Hossain, and Mohammed Moshiul Hoque. 2025. CUET-NLP_Big_O@DravidianLangTech 2025: A multimodal fusion-based approach for identifying misogyny memes. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 427–434, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*.

Javier Huertas-Tato, Christos Koutlis, Symeon Papadopoulos, David Camacho, and Ioannis Kompatsiaris. 2024. A clip-based siamese approach for meme classification. *Preprint*, arXiv:2409.05772.

Douwe Kiela et al. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Advances in Neural Information Processing Systems*.

Jiaqi Liu, Ran Tong, Aowei Shen, Shuzheng Li, Changlin Yang, and Lisha Xu. 2025. Memeblip2: A novel lightweight multimodal system to detect harmful memes. *Preprint*, arXiv:2504.21226.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Jiasen Lu et al. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*.

Shraman Pramanick, Shivam Sharma, Dimitar Dimitrov, Md Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021. Momenta: A multimodal framework for detecting harmful memes and their targets. *Preprint*, arXiv:2109.05184.

Saranya Rajiakodi, Shunmuga Priya Muthusamy Chinnan, Premjith B, Subalalitha CN, Rahul Ponnusamy, Anshid K A, Bhuvaneshwari Sivagnanam, Jananayagan V, Bharathi Raja Chakravarthi, Ragavan N, and Santhini P. 2026. Overview of the shared task on multilevel political meme classification in tamil and malayalam. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR.

A Error Analysis

Analysis of misclassifications reveals several consistent patterns across both tasks and languages. For Level-2 classification, models often confuse "Against individual person" and "Against party", as these categories share similar rhetorical expressions and visual cues in political memes. Performance also drops for minority classes, particularly "Support for individual person", which consistently receives much lower F1 scores than the dominant opposition categories.

This issue is especially evident in the Malayalam dataset, where the Level-1 test set contains only four Support/Praise instances compared to ninety-six Troll/Oppose samples. As a result, text-only models achieve high accuracy (96%) by predicting the majority class, while failing to detect the minority class, leading to zero F1 score for Support/Praise.

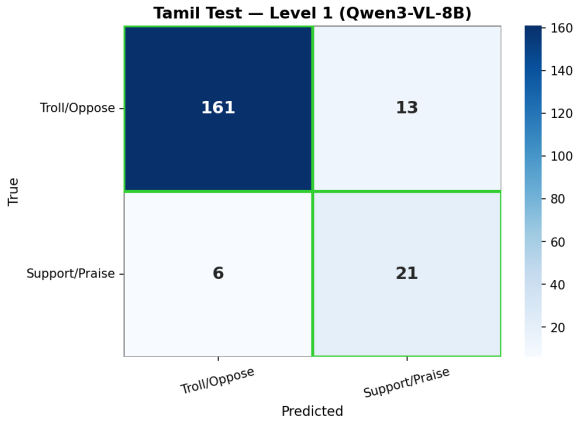


Figure 2: Tamil — Level 1 (Binary)

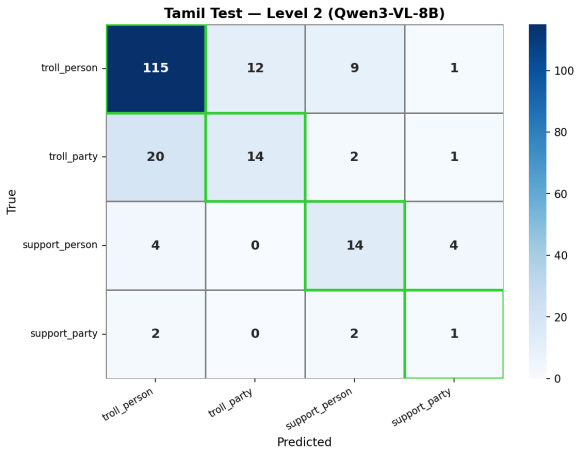


Figure 3: Tamil — Level 2 (4-class)

Visual information improves performance on the Tamil dataset by reducing confusion between fine-grained Level-2 categories. Vision-Language Models show better ability to distinguish person-oriented and party-oriented attacks. In the Tamil Level-1 matrix (Figure 2), the model correctly identifies the dominant Troll/Oppose class, while Level-2 (Figure 3) shows confusion between *troll_person* and *troll_party*. However, in the Malayalam dataset, severe class imbalance limits the ability of all models to learn reliable patterns for minority classes. The Malayalam Level-1 matrix (Figure 4) confirms near-total collapse onto the Troll/Oppose class, while Level-2 (Figure 5) shows that the *intersection* class is frequently misclassified as *troll_person*. These findings highlight the importance of balanced evaluation datasets and macro-averaged metrics.

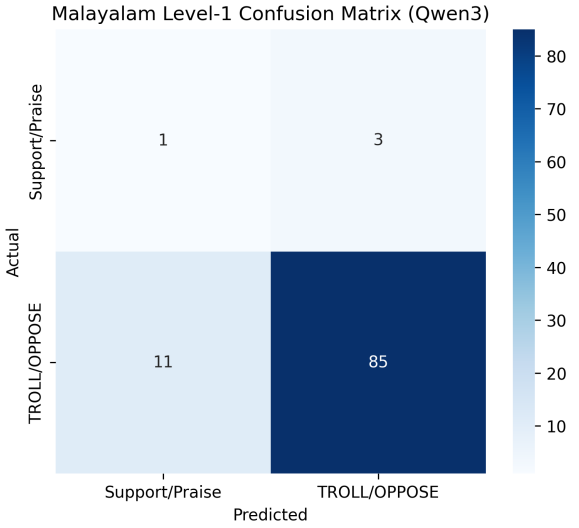


Figure 4: Malayalam — Level 1 (Binary)

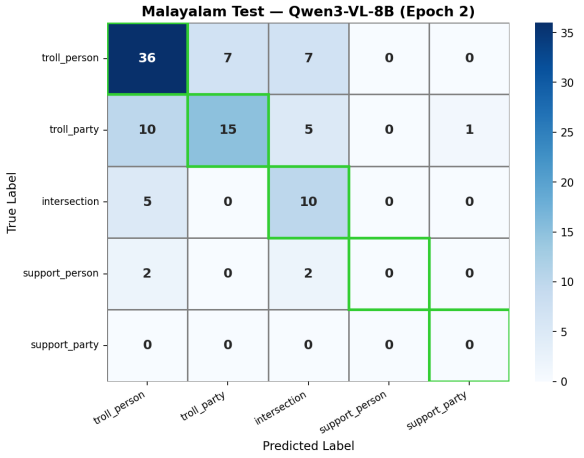


Figure 5: Malayalam — Level 2 (5-class)

SERENE@DravidianLangTech 2026: Multimodal Approaches for Depression Detection in Dravidian Speech: Acoustic, Spectrogram, and Transformer-Based Models

TT Pranesh, KK Thamizhmathi, S Vigneshwaran, B Bharathi
Department of Computer Science and Engineering
Sri Sivasubramania Nadar College of Engineering
pranesh2370060@ssn.edu.in
thamizhmathi2370055@ssn.edu.in
vigneshwaran2370061@ssn.edu.in
bharathib@ssn.edu.in

Abstract

This paper presents our submission to the Depression Detection in Dravidian Languages shared task at DravidianLangTech 2026. We investigate three complementary approaches for speech-based depression detection in Tamil and Malayalam: (i) acoustic feature engineering using MFCC and prosodic features with a Support Vector Machine (SVM) classifier, (ii) a convolutional neural network (CNN) trained on Mel-spectrogram representations, and (iii) a transformer-based model using Whisper-generated transcripts fine-tuned with XLM-RoBERTa. Experimental results show that acoustic feature-based SVM and spectrogram-based CNN models achieve the strongest performance on both Tamil and Malayalam datasets, while the transformer-based approach also produces competitive results. We further discuss limitations and future research directions.

1 Introduction

Depression is a prevalent mental health condition characterized by persistent sadness, reduced interest, cognitive impairment, and emotional instability. Recent research in computational linguistics and speech processing has explored automatic detection of depression using textual and acoustic cues. While significant progress has been made in high-resource languages such as English, low-resource languages such as Tamil and Malayalam remain underexplored.

The Depression Detection in Dravidian Languages shared task (G et al., 2026) aims to address this gap by providing speech data in Tamil and Malayalam for automatic depression classification. Speech signals contain rich acoustic and linguistic cues including prosody, lexical choice, hesitation patterns, and emotional tone.

In this work, we explore three modeling paradigms of increasing representational complexity:

- Acoustic feature-based classification using MFCC and prosodic features with SVM.
- Spectrogram image-based classification using a Convolutional Neural Network (CNN).
- Transformer-based textual classification using Whisper-generated transcripts fine-tuned with XLM-RoBERTa.

Our goal is to systematically compare traditional machine learning, deep convolutional models, and transformer-based approaches within the shared-task setting.

The remainder of this paper is organized as follows. Section 2 reviews existing literature on speech-based depression detection and related approaches. Section 3 describes the dataset characteristics and distribution across Tamil and Malayalam. Section 4 presents the proposed approaches, including acoustic feature-based classification, spectrogram-based convolutional neural networks, and transformer-based text modeling. Section 5 reports the experimental results and comparative performance analysis of the proposed models. Section 6 discusses the limitations of the current study. Finally, Section 7 concludes the paper and outlines potential directions for future research.

2 Related Work

Automatic depression detection has attracted growing attention in speech processing and natural language processing communities. Early research primarily relied on handcrafted acoustic features such as MFCCs, pitch, and energy statistics to model depression-related vocal characteristics. With the

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 354–358
July 3, 2026 ©2026 Association for Computational Linguistics

advancement of deep learning, convolutional neural networks (CNNs) operating on spectrogram representations became increasingly popular. Ensemble CNN frameworks for spectrogram-based depression detection have demonstrated improved robustness over classical machine learning approaches (Vázquez-Romero and Gallardo-Antolín, 2020). Deep CNN models applied to smartphone-recorded speech signals have further shown that learned spectral representations effectively capture depression-related acoustic patterns (Kim et al., 2023). Hybrid approaches combining MFCC-based handcrafted features with CNN-generated spectrogram representations highlight the complementary strengths of statistical and learned features (Das and Naskar, 2024). Recent work in Dravidian languages has also explored speech-based depression classification using deep learning techniques (Kritika et al., 2025). Previous shared task initiatives in Dravidian languages have also promoted research on multimodal and speech-based analysis of low-resource languages (Premjith et al., 2024).

More recently, transformer-based models have significantly advanced depression detection research. Hybrid CNN and transformer-based pretrained language models have reported notable improvements compared to traditional methods (Ramalakshmi et al., 2024). The emergence of large-scale foundation models has further strengthened text-based analysis. A multilingual automatic speech recognition system trained with large-scale weak supervision enables reliable transcription of low-resource languages (Radford et al., 2023). Speech and text transformer architectures have demonstrated strong cross-lingual generalization capabilities for depression detection across tasks and languages (Gómez-Zaragozá et al., 2025).

Beyond unimodal approaches, multimodal fusion techniques have also been investigated to integrate acoustic, textual, and behavioral signals. Multimodal data fusion has been shown to enhance robustness and improve generalization in depression detection systems (Nykoniuk et al., 2025). Multimodal modeling in Dravidian languages has also been explored for related tasks (Anilkumar et al., 2026). While prior work has largely focused on high-resource languages and multimodal datasets, systematic evaluation of acoustic, spectrogram-based, and transformer-driven methods in low-resource Dravidian languages remains limited. Our work addresses this gap by comparatively analyzing these paradigms for Tamil and Malayalam speech.

Table 1: Dataset statistics for Malayalam and Tamil.

Statistic	Malayalam	Tamil
Total Samples	1,888	1,534
Depressed Samples	888	534
Non-Depressed Samples	1,000	1,000
Unique Speakers	8	9
Depressed Speakers	3	4
Non-Depressed Speakers	5	5
Training Samples	1,688	1,374
Test Samples	200	160
Train (%)	89.41	89.57
Test (%)	10.59	10.43

3 Dataset Description

The dataset consists of speech recordings in two Dravidian languages, Tamil and Malayalam, where each utterance is labeled as *Depressed* or *Non-depressed*, forming a binary classification task. Speech signals contain indicators of depression through acoustic and linguistic cues such as prosody, speech rate, and lexical patterns. The corpus includes 1,888 Malayalam and 1,534 Tamil samples collected from 8 Malayalam speakers (3 depressed, 5 non-depressed) and 9 Tamil speakers (4 depressed, 5 non-depressed). The data is split into training and test sets with approximately 90% used for training and 10% for testing, as summarized in Table 1. Since multiple utterances are recorded from the same speakers, intra-speaker similarities may influence model performance.

4 Methodology

4.1 Acoustic Feature-Based Classification

In the first approach, handcrafted acoustic features are extracted using the Librosa library. A 33-dimensional feature vector is constructed from MFCC coefficients, pitch statistics, energy statistics, and silence ratio, capturing spectral, prosodic, and temporal characteristics of speech associated with emotional and psychological states. The extracted features include 13 MFCC coefficients summarized using mean and standard deviation, along with pitch and energy statistics and silence ratio.

The feature vectors are used to train a Support Vector Machine (SVM) classifier with a radial basis function (RBF) kernel. The model is trained with $C = 10$, $\gamma =$ scale, and balanced class weights to address class imbalance. The overall acoustic feature-based pipeline is illustrated in Figure 1.

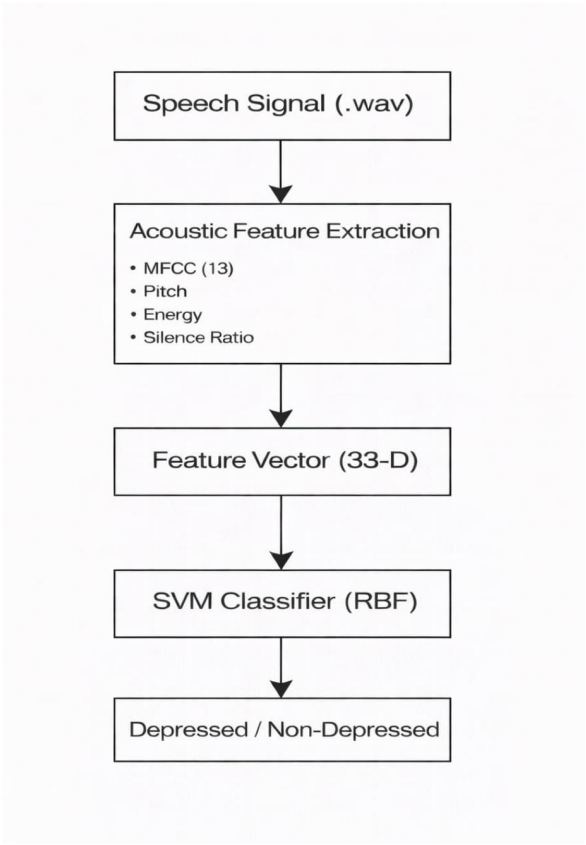


Figure 1: Acoustic feature-based depression detection pipeline

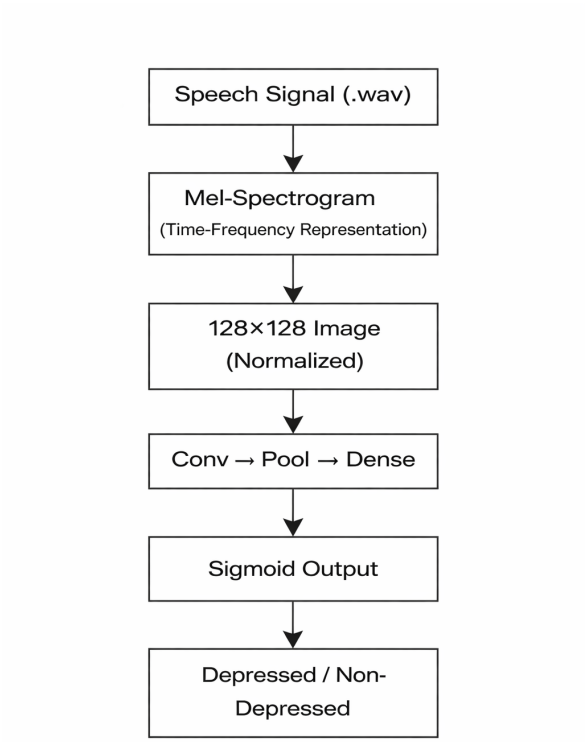


Figure 2: Spectrogram-based CNN pipeline.

4.2 Spectrogram-Based CNN Model

In the second approach, each speech signal is converted into a Mel-spectrogram representation using the Librosa library. Audio recordings are truncated to 5 seconds and transformed into Mel-frequency spectrograms in decibel (dB) scale to capture temporal and spectral speech characteristics.

The generated spectrograms are resized to 128×128 pixels and normalized before being used as input to a Convolutional Neural Network (CNN). The CNN architecture consists of two convolutional layers with 32 and 64 filters followed by max-pooling, a fully connected layer with dropout regularization, and a sigmoid output layer for binary classification.

The model is trained using binary cross-entropy loss and the Adam optimizer for 20 epochs. The overall spectrogram-based classification pipeline is illustrated in Figure 2.

4.3 Transformer-Based Text Classification

In the third approach, we incorporate linguistic information by converting speech signals into textual transcripts using the Whisper medium automatic speech recognition (ASR) model. Whisper is a multilingual transformer-based ASR system capable of transcribing low-resource languages such as Tamil.

The generated transcripts are cleaned by removing unnecessary punctuation and normalizing whitespace. The processed text is tokenized using the XLM-RoBERTa tokenizer and converted into subword embeddings.

We fine-tune XLM-RoBERTa-base, a multilingual transformer pretrained on 100 languages, for binary depression classification. All transformer encoder layers are updated during training, and the contextual representation of the classification token is passed through a linear classification head for prediction.

The model is trained using cross-entropy loss with a learning rate of 2×10^{-5}, batch size of 8, and 6 training epochs. The overall transformer-based pipeline is illustrated in Figure 3.

5 Results

We evaluate the proposed approaches using Accuracy(Acc), Macro-Precision(M-P), Macro-Recall(M-R), and Macro-F1(M-F1) score. Tables 2

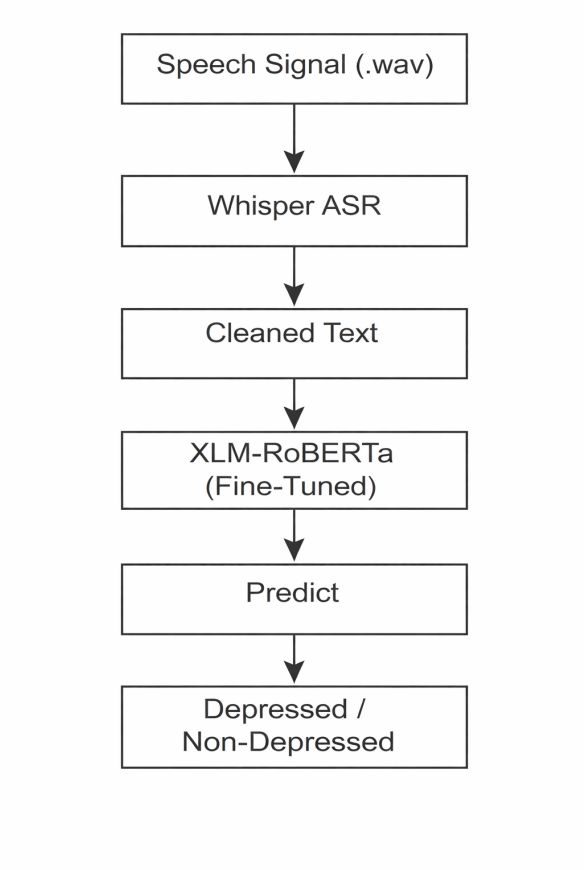


Figure 3: Transformer-based depression detection pipeline.

Table 2: Performance comparison on the Tamil dataset.

Model	Acc	M-P	M-R	M-F1
Acoustic + SVM	1.00	1.00	1.00	1.00
Spectrogram + CNN	1.00	1.00	1.00	1.00
Whisper + XLM-R	0.96	0.96	0.96	0.96

and 3 summarize the performance of the models on Tamil and Malayalam datasets. For Tamil, the acoustic feature-based SVM and spectrogram-based CNN models achieve near-perfect classification performance, while the transformer-based Whisper + XLM-RoBERTa approach obtains slightly lower scores. Similarly, on the Malayalam dataset, both acoustic and spectrogram-based models achieve near-perfect performance across all evaluation metrics. Our acoustic feature-based SVM system (Run 1) achieved **Rank 1** in both Tamil and Malayalam tracks of the shared task leaderboard. The implementation and experimental codes are publicly available on GitHub[1].

[1] https://github.com/Pranesh4950/Depression-Detection-Task

Table 3: Performance comparison on the Malayalam dataset.

Model	Acc	M-P	M-R	M-F1
Acoustic + SVM	1.00	1.00	1.00	1.00
Spectrogram + CNN	1.00	1.00	1.00	1.00

6 Limitations

The dataset size is relatively small and may limit generalization. Since multiple utterances are recorded from the same speakers, speaker-dependent patterns may influence model performance and lead to overestimated results. In addition, depressed and non-depressed recordings differ in recording conditions and sampling rates, which may introduce unintended acoustic cues that models can exploit. Whisper transcription errors may also affect transformer-based textual modeling. Future work will focus on speaker-independent evaluation and more robust validation settings.

7 Conclusion

We presented three approaches for depression detection in Tamil and Malayalam speech. Acoustic feature-based SVM and spectrogram-based CNN models achieved the strongest performance, while the transformer-based textual approach also produced competitive results. These findings highlight the effectiveness of acoustic and spectrogram representations for depression detection in speech. Due to the limited number of speakers and repeated utterances per speaker, speaker-independent evaluation remains an important direction for future work

8 Ethical Considerations

Depression detection systems should be used only as assistive tools and not as clinical diagnostic systems. Sensitive speech data must be handled responsibly with proper privacy protection and informed consent.

Acknowledgment of Generative AI Usage

Generative AI tools were used for language refinement and formatting assistance during manuscript preparation. All experiments and evaluations were conducted by the authors.

References

A. Anilkumar, Jyothish Lal, B. Premjith, and Bharathi Raja Chakravarthi. 2026. Dravlangguard: A multimodal approach for hate speech detection in dravidian social media. In *Speech and Language Technologies for Low-Resource Languages*, volume 2656 of *Communications in Computer and Information Science*, Cham. Springer.

Arnab Kumar Das and Ruchira Naskar. 2024. A deep learning model for depression detection based on MFCC and CNN-generated spectrogram features. *Biomedical Signal Processing and Control*, 90:105898.

Jyothish Lal G, Premjith B, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Durairaj Thenmozhi, and Prasanna Kumar Kumaresan. 2026. Shared task on depression detection from malayalam and tamil speech data. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Lucía Gómez-Zaragozá, Javier Marín-Morales, Mariano Alcañiz, and Mohammad Soleymani. 2025. Speech and text foundation models for depression detection: Cross-task and cross-language evaluation. In *Proceedings of Interspeech 2025*, pages 5253–5257.

Ah Young Kim, Eun Hye Jang, Seung-Hwan Lee, Kwang-Yeon Choi, Jeon Gue Park, and Hyun-Chool Shin. 2023. Automatic depression detection using smartphone-based text-dependent speech signals: A deep convolutional neural network approach. *Journal of Medical Internet Research*, 25:e34474.

A. Kritika, S. Meenakshy, Arya Palackal Shijish, Riya Rajeev, and Jyothish Lal. 2025. Dravimood: Speech-based depression classification in dravidian languages using feature fusion and deep learning. In *Proceedings of the Fourth International Conference on Speech and Language Technologies for Low-Resource Languages (SPELLL 2025)*.

Mariia Nykoniuk, Oleh Basystiuk, Nataliya Shakhovska, and Nataliia Melnykova. 2025. Multimodal data fusion for depression detection approach. *Computation*, 13(1):9.

B. Premjith, G. Jyothish, V. Sowmya, and B. Bharathi. 2024. Findings of the shared task on multimodal social media data analysis in dravidian languages (msmda-dl)@ dravidianlangtech 2024. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 56–61.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 28492–28518. PMLR.

N. R. Ramalakshmi, Meghna Ganesh Kumar, and S. Raghavi. 2024. Automated depression detection using CNN and transformer-based pre-trained language models. In *Proceedings of the International Conference on Computer, Communication, and Signal Processing*, pages 42–53. Springer.

Adrián Vázquez-Romero and Ascensión Gallardo-Antolín. 2020. Automatic detection of depression in speech using ensemble convolutional neural networks. *Entropy*, 22(6):688.

SJM_MINDS@DravidianLangTech@ACL2026: Machine Learning Approaches for Hope Speech Detection in Code-Mixed Tulu

Hosahalli Lakshmaiah Shashirekha, Manjula Suvarna and Jayashree Krishna
Department of Computer Science, Mangalore University, Mangalore, India
{hlsrekha, manjulashivamu, jayashreekmu}@mangaloreuniversity.ac.in

Abstract

Hope speech detection is an important task in understanding emotionally constructive communication in online platforms, especially in low-resource and code-mixed languages. This paper describes our system submitted to the first shared task on Hope Speech Detection in Code-Mixed Tulu, organized by DravidianLangTech@ACL 2026. The shared task consists of two tasks: Task 1 - Coarse-Grained Hope Tone Classification and Task 2 - Fine-Grained Hope Type Classification, with the objective of detecting and classifying the tone and type of hope expressed in code-mixed Tulu texts. We experimented with Logistic Regression (LR) and Linear Support Vector Classifier (LinearSVC) - classical Machine Learning (ML) approaches, trained with Term Frequency and Inverse Document Frequency (TF-IDF) of word ngrams (n = 1, 2). For Task 1, we employed both models, whereas for Task 2, we employed only the LR model. Linear SVC obtained a macro F1-score of 0.51 in Task 1 and secured 4th rank, while the LR model obtained a macro F1-score of 0.37 in Task 2 and secured 5th rank. The results demonstrate that traditional ML approaches remain effective for low-resource code-mixed language scenarios.

1 Introduction

Digital technologies have enabled instant information sharing and interaction through social networks such as Facebook, Instagram, and YouTube. This has led to the increased volume of user-generated content in regional and code-mixed languages. Users are taking advantage of social media and are sharing hate speech, hope speech, in addition to other types of content. Detecting constructive emotional content, such as hope speech, is essential for promoting positive online interactions and reducing harmful communication.

Hope can be described as an openness of spirit toward the future—a desire, expectation, or wish for something to occur or be true—that significantly influences a person's mindset, emotions, behavior, and decision-making (Balouchzahi et al., 2023). It is a powerful force that motivates individuals to persist despite life's uncertainties (Kumaresan et al., 2023). Detecting hope speech focuses on identifying whether a piece of text expresses hope or not. However, detecting it manually is both tedious and time-consuming because of the rising number of users and the expanding volume of social media content (Divakaran et al., 2024).

While sentiment analysis and emotion detection have been extensively studied, hope speech detection in low-resource and code-mixed languages such as Tulu remains underexplored. The detection of hope speech has emerged as a significant task, driven by the need to recognize motivational expressions and goal-oriented behavior of users on social media platforms (Ramos et al., 2025). As there are no guidelines to create any post/comment on social media, users usually combine words and sub-words belonging to more than one language they know, leading to code-mixed text. Code-mixed Tulu text usually combines Tulu with Kannada and English words, introducing challenges such as spelling variations, transliteration inconsistencies, and informal grammar.

Hope Speech Detection in Code-Mixed Tulu Language shared task[1], organized by DravidianLangTech@ACL 2026, aims to promote research in low-resource and code-mixed Tulu language in the context of social media text (Thenmozhi et al., 2026). The primary objective of this task is to automatically identify and categorize hope-related expressions in user-generated content. The shared task consists of two tasks:

- **Task 1: Coarse-Grained Hope Tone Classification** - consists of four classes - *Encouraging*, *Discouraging*, *Uninvolved*, and *Blended*

[1]https://www.codabench.org/competitions/11328/

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 359–365
July 3, 2026 ©2026 Association for Computational Linguistics

Table 1: Sample text with their English Translation and Corresponding Label

Sample Code-mixed Text	English Translation	Classes
Nett comdey olu unndu ...?	Where's the comedy in this?	Blended Hope
Mast samaryd boka..benner batter	We had a guest after a very long time	Uninvolved
ಬಾರಿಷೊಲಿಕುಲತ್ಂಡ್	It was very nice	Encouraging Hope
Anchor n onji change malple	Kindly change the anchor	Discouraging Hope

Tone, with the goal to identify the overall emotional tone of a given social media post (tweet, comment, etc.) based on the presence, absence, or ambiguity of hope-related expressions.

- **Task 2: Fine-Grained Hope Type Classification** - consists of five classes - *Optimistic Hope*, *Realistic Hope*, *Inspiring Hope*, *Fading Hope*, and *Hopelessness*, with the goal of refining the classification by identifying specific types of hope expressed in the text.

Few representative samples from the shared task dataset (Thenmozhi et al., 2026) along with their English translations, are shown in Table 1.

To address the challenges of the shared task, in this paper, we describe our proposed system for hope speech detection in code-mixed Tulu. We explored classical ML models: LR and LinearSVC, trained with TF-IDF of words n-grams (n = 1, 2) for hope speech detection.

The rest of the paper is organized as follows: Section 2 presents the related work, Section 3 describes the methodology, Section 4 discusses the experiments, results and error analysis and Section 5 concludes the paper with future work.

2 Related Work

Compared to high-resource languages, low-resource languages face issues such as limited labeled data, high dialectal variation, and lack of pre-trained language models. Despite these limitations, hope speech detection in low-resource languages has gained attention with the increase in the number of social media users and also the increasing volume of user-generated text. A few studies relevant in this direction are discussed below:

Mesay Gemeda Yigezu et al. (Yigezu et al., 2023) focused on detecting hope speech in social media posts in English and Spanish as part of IberLEF 2023 shared task. Support Vector Machine (SMV)-based model trained with TF-IDF of words achieved macro F1-scores of 0.489 for English and 0.481 for Spanish. Aggarwal et al. (Aggarwal et al., 2023) examined ML methods for classifying English YouTube comments into *Hope Speech*, *Non-Hope Speech*, or *Neutral*, using the EACL-2021: Hope Speech Detection for Equality, Diversity, and Inclusion" shared task dataset. They identified label inconsistencies and relabeled the dataset, testing traditional models (Naïve Bayes, LR, SVM) and BERT. BERT achieved the best results, with an F1-score of 0.6966, and improved to 0.85 when the neutral class was excluded for binary classification.

Hegde et al. (Hegde et al., 2023b) addressed the identification of hope speech in English and code-mixed texts in Bulgarian, Spanish, and Hindi, in a shared task on Hope Speech Detection for Equality, Diversity, and Inclusion (LT-EDI) at RANLP 2023. The authors proposed models that combined transformer-based embeddings, including BERT and multilingual BERT, with TF-IDF character n-gram features, and employed LinearSVC for classification. Their approaches achieved strong performance, securing top ranks across multiple language tracks. Uzor et al. (Uzor et al., 2025) compared SVM, deep learning models, and transformer-based approaches for hope speech detection. SBERT achieved the highest accuracy in both binary (92.81%) and multiclass (84.29%) tasks, outperforming BERT, GPT-2, and traditional models. Overall, transformer models proved most effective at capturing nuanced categories, though challenges remain for underrepresented classes in low-resource settings.

Abiola et al. (Abiola et al., 2026) evaluated traditional ML models against fine-tuned transformer architectures for hope speech detection. The LaBSE (Language-agnostic BERT Sentence Embedding) model consistently outperformed the Ridge Classifier baseline, achieving macro-F1 scores of 0.861 (English), 0.868 (Spanish), 0.948 (Urdu), and 0.839 (German). These results show that while traditional models remain competitive, transformer models more effectively capture subtle semantic patterns

in hope speech. Similarly, Sharma et al. (Sharma et al., 2025) proposed an ensemble framework for low-resource languages (English, Kannada, Malayalam, Tamil), combining LSTM, mBERT, and XLM-RoBERTa. Their ensemble outperformed individual models, with weighted F1-scores of 0.93 (English), 0.74 (Kannada), 0.82 (Malayalam), and 0.60 (Tamil), highlighting the strength of ensemble methods for multilingual hope speech detection in low-resource settings.

Research on Tulu NLP is steadily expanding through shared tasks and resource development initiatives led by multiple researchers. These efforts—ranging from sentiment analysis to parallel corpus construction—are gradually building the foundation for more advanced low-resource language processing in Tulu.

DravidianLangTech@RANLP 2023 introduced sentiment analysis for code-mixed Tamil and Tulu, marking the first public-domain study of Tulu sentiment analysis, where systems were evaluated using macro F1-score (Hegde et al., 2023a). To advance neural machine translation for low-resource Dravidian languages, a Kannada–Tulu parallel corpus was developed, with transformer-based models outperforming traditional encoder–decoder approaches (Hegde and Shashirekha, 2023). LT-EDI@EACL 2024 addressed the detection of homophobia and transphobia in multilingual social media comments, including Tulu (Chakravarthi et al., 2024). Morphological segmentation studies for Kannada and Tulu further demonstrated the effectiveness of CRF-based models with syllable and character features for morpheme boundary detection (Hegde and Shashirekha, 2024). The shared task at DravidianLangTech@NAACL 2025 addressed multi-class sentiment classification using data from platforms such as YouTube and Twitter (Thenmozhi et al., 2025). Narayanan et al. (Narayanan and Aepli, 2024) developed an English–Tulu parallel corpus and machine translation system for the low-resource Tulu language.

While most of the work focuses on English, low-resource languages, and code-mixed texts in low-resource languages continue to face issues such as lexical sparsity, limited annotated data, imbalanced data, and lack of pretrained models. This motivated us to explore traditional ML models for hope speech detection in code-mixed Tulu language.

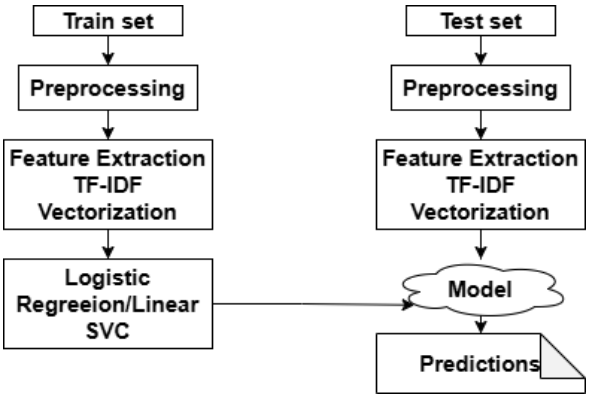


Figure 1: Framework of the Proposed Methodology for Hope Speech Detection in Code-mixed Tulu

3 Methodology

The workflow of the proposed methodology is depicted in Figure 1 and the steps involved are explained in the following subsections:

3.1 Preprocessing

Text data is subjected to a standardized preprocessing pipeline to ensure consistency and reduce noise. The preprocessing steps included lower-casing, removal of punctuation and special characters, and tokenization. These steps help normalize textual input and reduce sparsity in feature space, thereby improving the effectiveness of downstream tasks.

3.2 Feature Extraction

To transform textual data into numerical form, we extracted word n-grams (n = 1, 2) and vectorized them using TfidfVectorizer[2]. TF-IDF is widely used in text classification as it effectively captures the importance of terms relative to their frequency within individual documents and across the corpus.

3.3 Model Description

We employed the classical ML models - LR and LinearSVC, to detect hope speech in code-mixed Tulu text.

- **LR** classifier is a strong baseline for text classification due to its robustness, interpretability, and efficiency in high-dimensional feature spaces. LR model is trained with TF-IDF of word ngrams (n = 1, 2) to capture both individual words and short contextual dependencies with a maximum feature size of 5,000. Limiting the feature size helps control model complexity and reduce overfitting. To ensure

[2]https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

proper convergence during optimization, the maximum number of training iterations was set to 1000.

- **LinearSVC** aims to find an optimal hyperplane that maximizes the margin between classes. It is particularly effective for sparse text representations and often performs competitively in text classification tasks. LinearSVC is trained with TF-IDF of word ngrams (n = 1, 2) to capture both individual words and short contextual dependencies. The maximum feature size is set to 8,000 to allow a richer representation. Additionally, a minimum document frequency (min_df) of 2 is applied to remove rare terms that may introduce noise. Since the dataset exhibited class imbalance, we set the class_weight parameter to balanced. This automatically adjusts class weights inversely proportional to class frequencies, thereby mitigating bias toward majority classes and improving generalization on minority categories.

These classifiers are selected due to their effectiveness in handling high-dimensional sparse feature representations. We explored both LR and LinearSVC for Task 1, whereas only LR model for Task 2.

4 Experiments and Results

The dataset provided for the shared task consists of code-mixed Tulu comments collected from social media platforms. The distribution of data across Train and Development sets for Task 1 and 2 are shown in Tables 2 and 3, respectively. The Test sets for Task 1 and 2 consists of 1,284 and 682 samples respectively.

Model performance is evaluated by the organizers using Precision, Recall, and F1-score, and the models are ranked based on macro F1-score. The performances of the proposed models on Development and Test sets are reported in Tables 4 and 5, respectively. In Task 1, Linear SVC achieved a macro F1-score of 0.51 and secured 4th rank in the shared task, whereas for Task 2, LR model obtained a macro F1-score of 0.37 and secured 5th rank. The comparison of macro F1-scores of the participating teams for Task 1 and 2 are shown in Figures 2 and 3, respectively.

Table 2: Distribution of data in Train and Development Sets for Task 1

Class	Train Set	Development Set
Encouraging	1,895	406
Discouraging	711	153
Uninvolved	2,490	534
Blended Tone	895	191

Table 3: Distribution of data in Train and Development Sets for Task 2

Class	Train Set	Development Set
Optimistic Hope	380	81
Realistic Hope	503	108
Inspiring Hope	1,129	242
Fading Hope	236	51
Hopelessness	937	200

4.1 Error Analysis

To gain deeper insights into the classification behavior of the proposed models, the confusion matrices for Task 1 and Task 2 shown in Figures 4 and 5, are analyzed.

For Task 1, the confusion matrix shows that the model is able to correctly identify a significant portion of the *Encouraging* and *Uninvolved* categories However, the model fails to identify *Discouraging* and *Blended Tone*, as these categories often contain overlapping linguistic expressions that combine both supportive and discouraging sentiments. Additionally, several *Encouraging* and *Discouraging* comments are also incorrectly predicted as *Uninvolved*. This may be due to implicit sentiment expressions in code-mixed Tulu comments, where supportive or discouraging intent is not always explicitly conveyed by strong sentiment indicators.

For Task 2, the model demonstrates strong performance in identifying the *Inspiring Hope* and *Hopelessness* categories. However, the instances of *Optimistic Hope* and *Realistic Hope* are frequently misclassified as *Inspiring Hope*. This suggests that the classifier tends to associate general positive sentiment with inspirational expressions. Furthermore, the *Fading Hope* category shows very few correct predictions, indicating that the model struggles to capture the subtle linguistic cues of gradually diminishing hope. These misclassifications highlight the complexity of fine-grained hope classification, particularly in code-mixed social media text, where contextual and semantic nuances are crucial.

The relatively small dataset and the presence of

Table 4: Results for Development Sets of Task 1 and Task 2

Task	Model	Precision	Recall	F1-score	Accuracy
Task 1	LR	0.62	0.51	0.52	0.67
	LinearSVC	0.53	0.53	0.53	0.62
Task 2	LR	0.32	0.34	0.31	0.50

Table 5: Results for Test Sets of Task 1 and Task 2

Task	Model	Precision	Recall	F1-score	Accuracy	Rank
Task 1	LinearSVC	0.52	0.51	0.51	0.62	4
Task 2	LR	0.64	0.39	0.37	0.56	5

code-mixed Tulu text with informal spellings and linguistic variations, collectively contribute to the lower macro F1-score achieved by the models. Few misclassified instances are shown in Table 6.

5 Conclusion

This paper presents the models submitted for Hope Speech Detection in the Code-Mixed Tulu shared task organized by Dravidian-LangTech@ACL 2026. We experimented with LR and LinearSVC models using word n-grams (n = 1, 2) features. Linear SVC with class-weight balance achieved the macro F1-score of 0.51 in Task 1 and secured the 4th rank, while the LR model achieved the macro F1-score of 0.37 and secured the 5th rank in Task 2. The findings demonstrate that classical ML approaches remain competitive for low-resource and code-mixed language classification tasks. More feature engineering and other ML models will be explored further.

Acknowledgments

The authors acknowledge the use of OpenAI's ChatGPT for assistance in grammar checking, language refinement, and improving the presentation of the manuscript. All research contributions, interpretations, and conclusions are developed by the authors.

Limitations

Despite the promising results obtained in this study, several limitations remain.

- The dataset used for the experiments is relatively small, which may limit the model's ability to learn diverse linguistic patterns and generalize effectively to unseen data.
- The presence of code-mixed Tulu text introduces additional challenges due to informal spellings, transliteration variations, and inconsistent grammatical structures commonly found in social media comments. These characteristics make it difficult for traditional ML models to capture the full contextual meaning of the text.
- Class imbalance across categories, particularly in Task 2, where some classes, such as *Fading Hope* have significantly fewer instances. This imbalance may negatively affect the model's ability to accurately learn and predict minority classes.
- The use of TF-IDF for feature representation fails to capture deeper semantic relationships in the text.

Overall, the study is limited by the small dataset size, code-mixed text complexities, class imbalance and the limitations of the ML models to generalize.

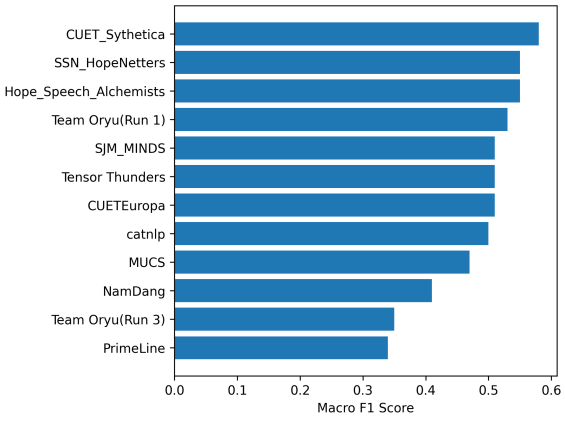


Figure 2: Comparison of Macro F1 scores of the Participating Teams for Task 1

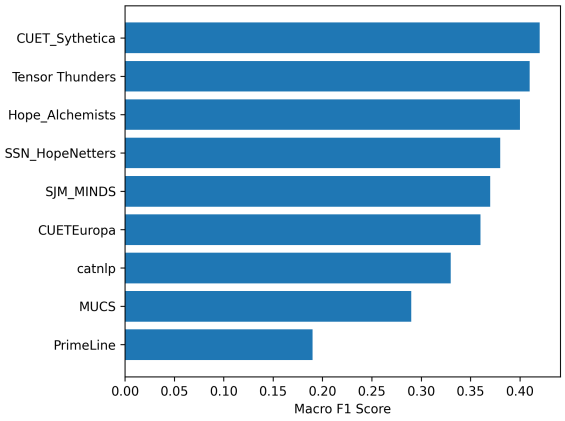


Figure 3: Comparison of Macro F1-scores of the Participating Teams for Task 2

Table 6: Sample Misclassified Instances in Task 1 and Task 2 along with their True and Predicted Labels

Task	Text	True	Predicted
Task 1	Ini pura picture baruva	Blended Tone	Uninvolved
	ಸತ್ಯೋದ ಬೆರಿಯೆ ಸತ್ಯೋಲು ಉಲ್ಲಾ ಪಂಡ್ ತೋಜಾದ್ ಕೊರಿಯರ್.	Encouraging	Uninvolved
	20 ne sala thupuni	Discouraging	Uninvolved
Task 2	ಅಜ್ಜ ಭಕ್ತಿ ಉಂಡು	Optimistic	Inspiring
	Tulunaad n oripuga... tulu bhashen oripaaga.	Realistic	Inspiring
	Yakshaganada choukattuve gotthijji.	Fading	Hopelessness
	ಈಈಈಈರ್ ಭಯಂಕರ ಮಾರ್ಡ್ 😔😔😁😁😉😉🤔🤔🤔	Inspiring	Hopelessness

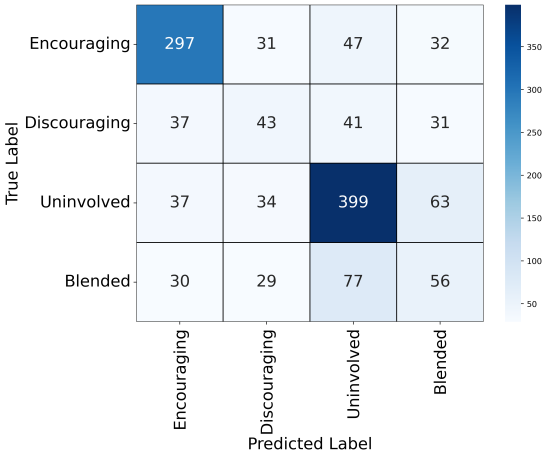


Figure 4: Confusion matrix for Task 1 - LinearSVC Model

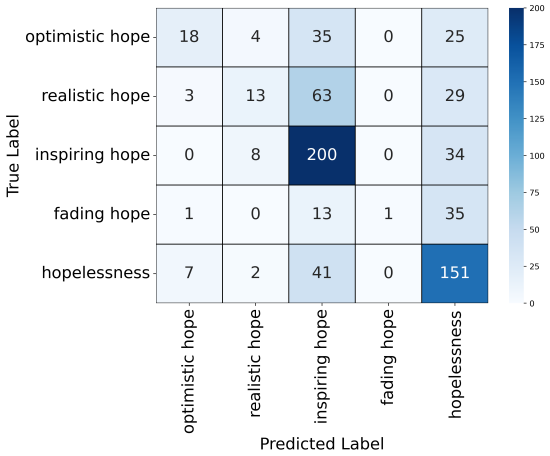


Figure 5: Confusion matrix for Task 2 - LR Model

References

Oluwatobi Joseph Abiola, Tolulope Olalekan Abiola, Obdulia Pichardo Lagunas, and Grigori Sidorov. 2026. Hope Speech Detection: A Comparative Study across Four Languages. *International Journal of Combinatorial Optimization Problems and Informatics*, 17(2):215–230.

Pranjal Aggarwal, Pasupuleti Chandana, Jagrut Nemade, Shubham Sharma, Sunil Saumya, and Shankar Biradar. 2023. Hope Speech Detection on Social Media Platforms. In *Cybercrime in Social Media*, pages 67–84. Chapman and Hall/CRC.

Fazlourrahman Balouchzahi, Grigori Sidorov, and Alexander Gelbukh. 2023. Polyhope: Two-Level Hope Speech Detection from Tweets. *Expert Systems with Applications*, 225:120078.

Bharathi Raja Chakravarthi, Prasanna Kumar Kumaresan, Ruba Priyadharshini, Paul Buitelaar, Asha Hegde, Hosahalli Lakshmaiah Shashirekha, Saranya Rajiakodi, Miguel Ángel García-Cumbreras, Salud María Jiménez-Zafra, José Antonio García-Díaz, Rafael Valencia-García, Kishore Kumar Ponnusamy, Poorvi Shetty, and Daniel García-Baena. 2024. Overview of Third Shared Task on Homophobia and Transphobia Detection in Social Media Comments. In *Proceedings of the Fourth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 124–132, St. Julian's, Malta. Association for Computational Linguistics.

S. Divakaran, K. Girish, and H. L. Shashirekha. 2024. Hope on the Horizon: Experiments with Learning Models for Hope Speech Detection in Spanish and English. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2024), Co-located with the 40th Conference of the Spanish Society for Natural Language Processing (SEPLN 2024)*. CEUR Workshop Proceedings.

Asha Hegde, Bharathi Raja Chakravarthi, Hosahalli Lakshmaiah Shashirekha, Rahul Ponnusamy, Subalalitha Chinnaudayar Navaneethakrishnan, Lavanya Sambath Kumar, Durairaj Thenmozhi, Martha Karunakar, Shreya Sriram, and Sarah Aymen. 2023a. Findings of the Shared Task on Sentiment Analysis in Tamil and Tulu Code-Mixed Text. pages 64–71, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Asha Hegde, Kavya G, Sharal Coelho, and Hosahalli Lakshmaiah Shashirekha. 2023b. MUCS@LT-EDI2023: Learning Approaches for Hope Speech Detection in Social Media Text. In *Proceedings of the Third Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 279–286, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Asha Hegde and Hosahalli Lakshmaiah Shashirekha. 2023. KT2: Kannada–Tulu Parallel Corpus Construction for Neural Machine Translation. In *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, pages 743–753, Goa University, Goa, India. NLP Association of India (NLPAI).

Asha Hegde and Hosahalli Lakshmaiah Shashirekha. 2024. *Syllable-Level Morphological Segmentation of Kannada and Tulu Words*, pages 113–133. Wiley, Hoboken, NJ, USA.

Prasanna Kumar Kumaresan, Bharathi Raja Chakravarthi, Subalalitha Cn, Miguel Ángel García-Cumbreras, Salud María Jiménez Zafra, José Antonio García-Díaz, Rafael Valencia-García, Momchil Hardalov, Ivan Koychev, Preslav Nakov, Daniel García-Baena, and Kishore Kumar Ponnusamy. 2023. Overview of the Shared Task on Hope Speech Detection for Equality, Diversity, and Inclusion. In *Proceedings of the Third Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 47–53, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Manu Narayanan and Noëmi Aepli. 2024. A Tulu Resource for Machine Translation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1756–1767, Torino, Italia. ELRA and ICCL.

Luis Ramos, Hiram Calvo, and Olga Kolesnikova. 2025. Hope Speech Detection in Social Media English Corpora: Performance of Traditional and Transformer Models. *Preprint*, arXiv:2510.23585.

Deepawali Sharma, Vedika Gupta, Vivek Kumar Singh, and Bharathi Raja Chakravarthi. 2025. Stop the Hate, Spread the Hope: An Ensemble Model for Hope Speech Detection in English and Dravidian Languages. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 24(3):1–18.

Durairaj Thenmozhi, Bharathi Raja Chakravarthi, Asha Hegde, Hosahalli Lakshmaiah Shashirekha, Rajeswari Natarajan, Sajeetha Thavareesan, Ratnasingam Sakuntharaj, Krishnakumari Kalyanasundaram, Charmathi Rajkumar, Poorvi Shetty, and Harshitha S Kumar. 2025. Overview of the Shared Task on Sentiment Analysis in Tamil and Tulu. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 732–738, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Durairaj Thenmozhi, Rathnakar Shetty P, Parameshwar R. Hegde, M D Anusha, Raksha Adyanthaya, Mohammed Fadhel Aljunid, Prasanna Kumar Kumaresan, and Bharathi Raja Chakravarthi. 2026. Findings of the Shared Task on Hope Speech Detection in Tulu. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 45–55, Melbourne, Australia. Association for Computational Linguistics.

GodsGift Uzor, Mohammadsaleh Pajuhanfard, Michael Beebe, Maaz Amjad, and Victor Sheng. 2025. Hope Speech Detection: A Comparative Study of Machine Learning, Deep Learning, and Transformer-Based Models. *Journal of Artificial Intelligence Research*, 74:123–145.

Mesay Gemeda Yigezu, Girma Yohannis Bade, Olga Kolesnikova, Grigori Sidorov, and Alexander F Gelbukh. 2023. Multilingual Hope Speech Detection Using Machine Learning. In *Proceedings of the IberLEF 2023 Workshop at SEPLN*, volume 3498, pages 123–134. CEUR Workshop Proceedings.

SSN_HopeNetters@DravidianLangTech 2026: Multi-Level Hope Speech Detection using XLM-RoBERTa

Moogambigai A, Nandhika Saravanan, Nikhil Karthik S, Pandiarajan D, Bharathi B
Department of Computer Science and Engineering
Sri Sivasubramaniya Nadar College of Engineering, Chennai, India
moogambigai2370071@ssn.edu.in
nandhika2370028@ssn.edu.in
nikhilkarthik2370024@ssn.edu.in
pandiarajan2370062@ssn.edu.in
bharathib@ssn.edu.in

Abstract

This paper presents our system submission to the Shared Task on Hope Speech Detection in Code-Mixed Tulu Language at DravidianLangTech @ ACL 2026. We introduce a transformer-based approach built on XLM-RoBERTa-base for multilingual hope speech classification. Our system addresses two subtasks: coarse-grained classification of hope versus non-hope speech and fine-grained categorization of different hope expressions. Since hope is often expressed in subtle ways, especially in mixed-language text, our model looks at the full context of a sentence to understand its real meaning rather than just focusing on specific words. Experimental results demonstrate that multilingual transformer models effectively model supportive and encouraging expressions, underscoring their suitability for promoting constructive discourse in low-resource and code-mixed language settings.

1 Introduction

The rapid growth of social media platforms (Chakravarthi, 2022) has transformed the way people express opinions, emotions, and social support online. While significant research has focused on identifying harmful and hateful content, comparatively less attention has been given to detecting positive and encouraging expressions such as hope speech. Identifying such content can help promote inclusivity and constructive communication in online communities.

In this work, we present SSN_HopeNetters, a transformer-based system for hope speech detection in code-mixed social media text. The task is challenging due to multilingual usage, code-mixing, spelling variations, and contextual ambiguity. To address these challenges, we employ XLM-RoBERTa-base, which is effective for multilingual representation learning.

Our approach fine-tunes the pretrained model on the provided dataset using appropriate preprocessing and hyperparameter tuning. Experimental results demonstrate that transformer-based contextual embeddings effectively capture semantic cues associated with hopeful and supportive expressions.

Keywords

Hope Speech, Positive Language Modeling, Cross-lingual Representation Learning, Context-aware Text Understanding, Code-mixed Social Media Text, Inclusive Language Identification, Transformer Fine-tuning

2 Related Work

Research in text classification and predictive modeling has evolved through both traditional machine learning frameworks and task-specific linguistic modeling. Early work on learning predictive structures from multiple tasks and unlabeled data demonstrated the importance of leveraging shared representations for improved generalization(Ando and Zhang, 2005). Such frameworks laid the foundation for handling limited labeled data settings, which is particularly relevant for emerging tasks like hope speech detection.

Traditional approaches to text classification have influenced the development of systems for detecting socially meaningful content, including hope speech (Balouchzahi et al., 2021). Earlier statistical and log-linear modeling techniques demonstrated the feasibility of scalable learning in high-dimensional textual spaces, providing a foundation for later advances in discourse-level classification tasks (Andrew and Gao, 2007). Although hope speech detection involves more nuanced semantic interpretation, these foundational approaches contributed to the broader understanding of efficient text modeling strategies.

From a linguistic and structural standpoint, computational methods for string processing and sequence analysis have shaped modern text analy-

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 366–370
July 3, 2026 ©2026 Association for Computational Linguistics

sis techniques (Gusfield, 1997). Furthermore, advances in syntactic parsing, such as the Yara Parser (Rasooli and Tetreault, 2015), highlight the importance of structural modeling in understanding sentence-level relationships. Such structural insights are relevant for hope speech detection, where meaning often depends on contextual cues, compositional structure, and implicit positive intent.

The task of hope speech detection has been formally introduced in recent years, emphasizing the identification of positive, supportive, and inclusive language in online discourse (Chakravarthi, 2022). Unlike hate speech detection, hope speech identification requires modeling subtle contextual cues and positive intent, making it inherently challenging.

More recently, shared task evaluations have expanded hope speech detection to low-resource and underrepresented languages. The findings of the shared task on hope speech detection in Tulu highlight challenges such as limited annotated data, linguistic diversity, and class imbalance. These studies underscore the importance of developing robust models for multilingual and low-resource settings.

3 Dataset

The dataset used in the shared task consists of social media comments annotated for hope speech detection in code-mixed Tulu (Thenmozhi et al., 2026). The data contains multilingual user-generated content, with labels indicating whether a given comment expresses hope speech or not. The overall distribution of the dataset is presented in Table 1.

Training Data: The training set contains labeled comments with fields including *ID*, *TEXT*, and *LABEL*. These annotations distinguish between hope speech and non-hope speech instances.

Test Data: The test set includes comments with *ID* and *TEXT* fields, without labels. Model predictions were generated on this unseen data for evaluation.

The dataset contains informal and user-generated text, introducing challenges such as spelling variations, code-mixing, transliteration, and contextual ambiguity. Class imbalance was observed between categories, making the classification task more challenging.

Basic preprocessing was performed to standardize the input format, including label encoding and stratified splitting into training and validation subsets. This ensured compatibility with transformer-based tokenization and fine-tuning procedures.

Task	Language	Train	Dev	Test
Task 1 (Coarse-Grained)	Tulu	5252	1126	1126
Task 2 (Fine-Grained)	Tulu	3185	682	683

Table 1: Dataset statistics for the Hope Speech Detection shared task.

4 Methodology

Our approach to hope speech detection is based on fine-tuning a pretrained multilingual transformer model. The overall framework of the proposed system is illustrated in Figure 1.

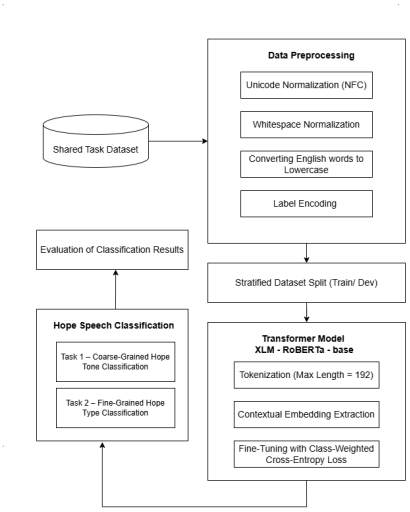


Figure 1: Framework of the Proposed Methodology

4.1 Model Architecture

XLM-RoBERTa-base:

We employed the pretrained XLM-RoBERTa-base model for multilingual text classification (Malik et al., 2023). XLM-RoBERTa is a transformer-based multilingual language model pretrained using masked language modeling on large-scale multilingual corpora. The model was fine-tuned on the labeled hope speech dataset to adapt its contextual representations to the classification task.

A task-specific classification head was added on top of the transformer encoder to predict the probability distribution over the output classes.

XLM-RoBERTa was selected due to its strong cross-lingual representation capability and its effectiveness in handling multilingual and code-mixed social media text (Zhao and Tao, 2021). Its subword tokenization mechanism also enables robust handling of transliterated and out-of-vocabulary words commonly found in informal online communication.

Hyperparameter	Value
Model	XLM-RoBERTa-base
Learning Rate	2e-5
Batch Size	16
Epochs	5
Optimizer	AdamW
Max Sequence Length	128
Loss Function	Cross-Entropy
Scheduler	Linear Warmup

Table 2: Training hyperparameters

4.2 Preprocessing

To ensure data consistency and effective model training, the following preprocessing steps were applied:

- **Text Normalization:** Basic cleaning was performed by removing unnecessary special characters and formatting inconsistencies while preserving semantic information.
- **Label Encoding:** Categorical labels were converted into numerical format for supervised learning.
- **Tokenization:** Text inputs were tokenized using the XLM-RoBERTa subword tokenizer, enabling effective handling of code-mixed and transliterated text.
- **Dataset Splitting:** The dataset was divided into training and validation subsets using stratified sampling to preserve class distribution.

4.3 Training Setup

The model was fine-tuned using cross-entropy loss for multi-class classification. Optimization was performed using the AdamW optimizer with learning rate scheduling to improve convergence stability.

Table 2 summarizes the training configuration used in our experiments.

Performance was evaluated using Accuracy, Precision, Recall, and Macro-F1 score on the validation set.

5 Experimental Results

This section presents the official leaderboard rankings and detailed evaluation results obtained by our team, **SSN_HopeNetters**, for both coarse-grained and fine-grained hope speech detection tasks.

5.1 Leaderboard Performance

Table 3 summarizes the official shared task leaderboard rankings.

5.2 Quantitative Analysis

Table 4 presents the validation performance of our proposed system for both tasks.

For Task 1, the model achieved an accuracy of 68.69% with a Macro-F1 score of 51.44%. Better performance was observed for majority classes such as *encouraging hope* and *uninvolved*, while lower performance on minority categories was mainly due to class imbalance.

For Task 2, the model achieved an accuracy of 44.43% and a Macro-F1 score of 38.52%. The lower performance reflects the increased difficulty of distinguishing fine-grained hope categories with overlapping semantic meanings.

Overall, the results indicate that multilingual contextual embeddings effectively capture hope-related expressions in code-mixed text, although fine-grained classification remains challenging.

5.3 Error Analysis

To better understand model behavior, we analyzed commonly occurring misclassification patterns observed in the validation predictions.

Error analysis revealed several challenges associated with code-mixed hope speech detection. The model frequently confused semantically similar hope categories due to overlapping emotional expressions. Additionally, transliterated spelling variations, noisy social media syntax, and implicit expressions of hope contributed to classification ambiguity.

Minority classes such as *fading hope* exhibited lower recall because of insufficient training examples. The findings indicate that fine-grained hope speech detection remains particularly difficult in low-resource multilingual settings.

5.4 Confusion Matrix Analysis

Figures 2 and 3 present the confusion matrices for Task 1 and Task 2.

The confusion matrices show that most errors occur between semantically related classes. In Task 2, minority categories such as *fading hope* experienced comparatively higher misclassification rates, highlighting the difficulty of fine-grained hope speech detection in low-resource code-mixed settings.

6 Conclusion

In this work, we presented a transformer-based approach for hope speech detection using the pre-

Task 1: Coarse-Grained Hope Tone Classification							
Team Name	Run	Acc	Macro-P	Macro-R	W-P	Macro-F1	Rank
SSN_HopeNetters	Run 1	0.64	0.56	0.56	0.68	0.55	2

Task 2: Fine-Grained Hope Type Classification							
Team Name	Run	Acc	Macro-P	Macro-R	W-P	Macro-F1	Rank
SSN_HopeNetters	Run 1	0.45	0.40	0.43	0.49	0.38	4

Table 3: Official leaderboard performance of SSN_HopeNetters

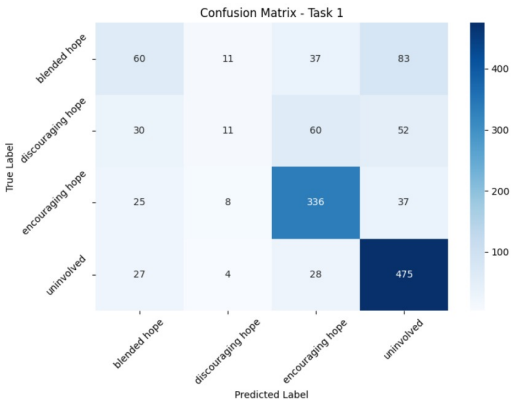


Figure 2: Task 1 Confusion Matrix

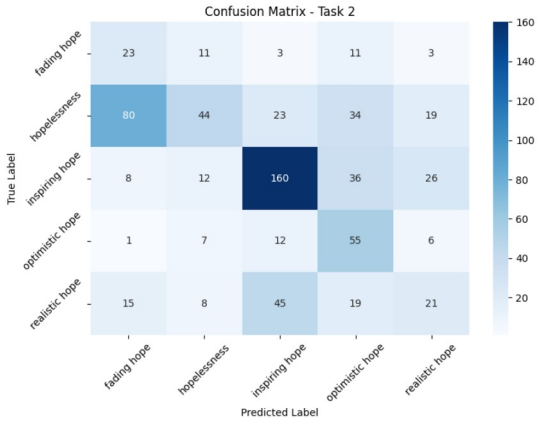


Figure 3: Task 2 Confusion Matrix

Task	Accuracy	Macro F1
Task 1 (Coarse)	0.6869	0.5144
Task 2 (Fine)	0.4443	0.3852

Table 4: Validation performance of SSN_HopeNetters

Actual	Predicted	Reason
Inspiring Hope	Optimistic Hope	Semantic overlap
Hope Speech	Non-Hope	Implicit positivity
Fading Hope	Uninvolved	Minority class sparsity

Table 5: Common misclassification patterns

trained XLM-RoBERTa model. By fine-tuning the model on labeled social media data, the system effectively learned contextual and semantic patterns associated with hopeful and supportive expressions. The multilingual pretraining of XLM-RoBERTa enabled robust handling of informal language, transliteration, and code-mixed content commonly found in user-generated text.

The experimental results demonstrate that contextualized transformer representations significantly enhance the ability to distinguish hope speech from non-hope speech. Overall, the study highlights the effectiveness of large-scale multilingual language models for socially relevant text classification tasks.[1]

7 Limitations

Despite achieving competitive performance, several limitations remain. First, the dataset size is relatively limited, which may restrict the model's ability to generalize to unseen domains or emerging linguistic patterns. Hope speech often depends on subtle contextual cues and implicit meaning, which can be challenging even for transformer-based models.

Additionally, the presence of informal language, spelling variations, and code-mixed expressions increases classification complexity. The model may occasionally misclassify borderline cases where hope is expressed indirectly or sarcastically. Future work should explore larger annotated datasets, improved contextual modeling strategies, and domain adaptation techniques to further enhance robustness and generalization.

[1]Code and implementation details are available at the project repository: GitHub Repository.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Fazlourrahman Balouchzahi, Aparna B K, and H L Shashirekha. 2021. MUCS@LT-EDI-EACL2021:CoHope-hope speech detection for equality, diversity, and inclusion in code-mixed texts. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 180–187, Kyiv. Association for Computational Linguistics.

Bharathi Raja Chakravarthi. 2022. Hope speech detection in youtube comments. *Social Network Analysis and Mining*, 12(1):75.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Muhammad Shahid Iqbal Malik, Anna Nazarova, Mona Mamdouh Jamjoom, and Dmitry I Ignatov. 2023. Multilingual hope speech detection: A robust framework using transfer learning of fine-tuning roberta model. *Journal of King Saud University-Computer and Information Sciences*, 35(8):101736.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

Durairaj Thenmozhi, Rathnakar Shetty P, Parameshwar R. Hegde, Anusha M D, Raksha Adyanthaya, Mohammed Fadhel Aljunid, Prasanna Kumar Kumaresan, and Bharathi Raja Chakravarthi. 2026. Findings of the Shared Task on Hope Speech Detection in Tulu. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Yingjia Zhao and Xin Tao. 2021. Zyj@ lt-edi-eacl2021: Xlm-roberta-based model with attention for hope speech detection. In *Proceedings of the first workshop on language technology for equality, diversity and inclusion*, pages 118–121.

Still Loading@DravidianLangTech 2026: Telugu Prompt-Style Recovery using Multilingual Transformers

Samonwita Sarker, Isnat Mehrin Sami, Priyontee Mojumder
Arpita Mallik, Hasan Murad
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology, Bangladesh
sarker.samonwita@gmail.com, u2104038@student.cuet.ac.bd,
u2104010@student.cuet.ac.bd, u2104012@student.cuet.ac.bd,
u2004023@student.cuet.ac.bd, hasanmurad@cuet.ac.bd

Abstract

This paper describes the system that our *Still Loading* team designed to run the Telugu Prompt-Style Recovery shared task at DravidianLangTech@ACL 2026. The purpose of the given task is categorizing Telugu transcript passages as belonging to one of 9 communicative styles: *Formal*, *Informal*, *Optimistic*, *Pessimistic*, *Humorous*, *Serious*, *Inspiring*, *Authoritative*, and *Persuasive*. We compared several multilingual Transformer-based models, i.e. MuRIL, XLM-RoBERTa-Large, mBERT, and IndicBERTv2. We chose a "Turbo Sandwich" preprocessing strategy which helps to give more emphasis to lexical deltas, in addition to Focal Loss. Our system based on the MuRIL was rated at the 7th place in the official leaderboard with a Macro-F1 rating of 0.1703, while post-submission experiments show that IndicBERTv2 achieves 0.3000 under identical conditions. The source code to reproduce our experiments is publicly available on Github[1]

1 Introduction

The most recent engineering advancement in Transformer architecture platforms has brought about incredible strides in natural language understanding (NLU). The languages with low resources, such as the Telugu language, are often paralyzed by the lack of properly annotated data on more complex tasks, such as style classification, which makes it difficult to develop an effective computational model. The Telugu Prompt-Style Recovery shared task, planned by DravidianLangTech@ACL2026, aims to fill this gap, providing a set of standard data and a reference point when it comes to the systematic determination of communicative styles in Telugu text.(DravidianLangTech Organizing Committee, 2024).

[1]Still-Loading-Prompt-Recovery-for-LLM-in-Telugu:https://github.com/Priyontee1713/Still-Loading-Prompt-Recovery-for-LLM-in-Telugu

The categorization of the stylistic overtones requires a representation of subtle tonal heuristics and pragmatic expectations, which are invaluable in personalized conversational agents and tone-uniform generation of prompts, to content regulation. The assignment outlines nine stylistic classes including *Formal*, *Informal*, *Optimistic*, *Pessimistic*, *Humorous*, *Serious*, *Inspiring*, *Authoritative* and *Persuasive*, each portraying a different tonal or pragmatic focus.

Label ambiguity between similar styles (e.g., *Pessimistic* and *Serious*), the small corpus of annotations ($3,000$ training instances) and the agglutinative morphology of Telugu and code-mixing of the language are some of the fundamental problems involved.

To find our way through these barriers, we do a comparative analysis of multilingual Transformer architectures, which finally leads to the choice of the leading model, which is MuRIL. The main achievements of this research project are as follows:

- A comparative study of multilingual Transformer-based models on Telugu prompt-style classification, and finding IndicBERTv2 are the best for this task.
- A new "Turbo Sandwich" input approach that exploits sequence matching to identify and evaluate lexical differences between original and modified text, allowing the encoder to focus on stylistic edits.
- Focal Loss with square-root weight dampening to encourage equal learning of the nine stylistic categories, dampening the impact of confusable class pairs.

2 Related Work

Telugu NLP has progressed from rule-based systems to Transformer-based architectures. Vemula et al. (2022) introduced TeQuAD, highlighting data

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 371–375
July 3, 2026 ©2026 Association for Computational Linguistics

scarcity in Dravidian languages, while sub-word modelling has shown value for morphologically rich languages (Bhattacharyya and Bhattacharya, 2025) and multilingual pretraining has benefited offensive language detection (B et al., 2024). Models such as XLM-RoBERTa (Conneau et al., 2020), MuRIL (Khanuja et al., 2021), and IndicBERTv2 (Kakwani et al., 2020) bridge the resource gap through cross-lingual and Indic-specific pretraining, with prompt-based fine-tuning showing further promise (Ullah et al., 2025).

Prompt engineering and recovery have become active research areas. Carlini et al. (2021) showed that inputs can be recovered from model outputs, while Lester et al. (2021) demonstrated prompt tuning as an effective low-resource strategy. Surveys by Sahoo et al. (2024) and Vatsal et al. (2025) catalogue techniques across multilingual settings, and Lewis et al. (2020) illustrates how attending to specific input segments improves task-focused classification, a principle central to our Turbo Sandwich strategy.

3 Data

This paper uses the Telugu language dataset provided by the Shared Task on Prompt Recovery at DravidianLangTech@ACL 2026 (DravidianLangTech Organizing Committee, 2024). The dataset includes Telugu transcript excerpts consisting of an original transcript and its stylistically modified version, each annotated with a target communicative style. The data are partitioned into training, development and test sets with 3,000, 300 and 301 samples respectively, with near-uniform class balance across all nine styles (321–347 instances per class; see Table 1).

Style	Train	Dev	Test
Formal	327	36	38
Informal	321	47	35
Optimistic	331	29	33
Pessimistic	347	29	22
Humorous	344	27	31
Serious	324	35	47
Inspiring	332	33	37
Authoritative	338	33	30
Persuasive	336	31	28
Total	**3,000**	**300**	**301**

Table 1: Label distribution across the training, development, and test splits.

4 Methodology

We build our system upon the MuRIL (Multilingual Representations for Indian Languages) model for sequence classification. An overview of hyperparameters is provided in Table 2.

Hyperparameter	Value
Base Model	MuRIL (*base-cased*)
Max Sequence Length	256
Batch Size	8
Learning Rate	4e-5
Optimizer	AdamW
Scheduler	Cosine
Warmup Ratio	0.1
Weight Decay	0.05
Epochs	15
Focal Loss (γ)	3
FP16 Precision	True

Table 2: Hyperparameter configuration for the MuRIL training pipeline.

4.1 Data Pre-processing: Turbo Sandwich Strategy

We **propose** the "Turbo Sandwich" strategy as a **novel** preprocessing formulation specifically designed to isolate stylistic changes between an original and a modified text. Rather than feeding the full modified text directly to the encoder, our pipeline applies a word-level sequence-matching approach using the `difflib` library to identify added or substituted tokens. As illustrated in Figure 1, these lexical deltas are then formatted alongside their immediate context, constructing the input as:

`Delta: {added_text} | Context: {context_text}`

This representation forces the Transformer encoder to attend specifically to the stylistic edits rather than treating the full sequence uniformly. To validate the contribution of this strategy, we compare it against a plain baseline (concatenating original and modified text with a separator), as reported in Table 4.

4.2 MuRIL Architecture

We optimize the MuRIL (`google/muril-base-cased`) model for sequence classification. Pre-trained on 17 Indian languages with millions of translations and transliterations, MuRIL captures phonetic, morphological, and script-level peculiarities of Telugu, enabling a linear classification head over the pooled representation to learn the 9 styles.

4.3 Training Configuration and Objectives

The training hyperparameters are summarized in Table 2. We employ AdamW with mixed-precision

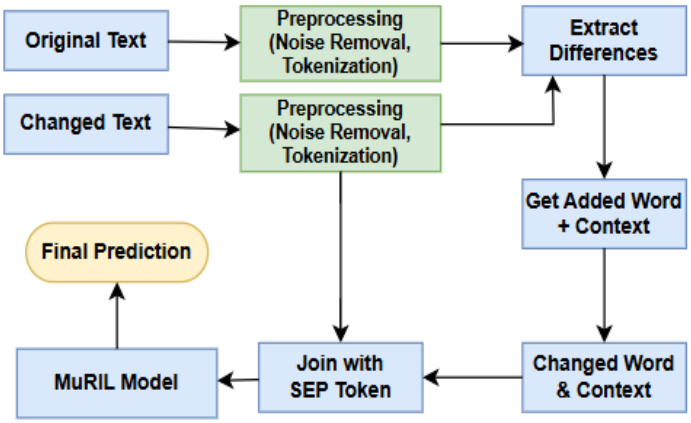


Figure 1: The Turbo Sandwich pre-processing strategy. The sequence matcher identifies stylistic deltas which are then formatted to highlight specific lexical changes for the MuRIL encoder.

training (FP16) for computational efficiency.

Instead of standard cross-entropy, we utilize Focal Loss (Lin et al., 2017) to address the "hard" misclassifications between similar styles. Given a model's predicted probability p_t for the true class, the focal loss is defined as:

$$\mathcal{L}_{\text{FL}} = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \qquad (1)$$

where $\gamma = 3$ is the focusing parameter and α_t represents the per-class weight. We smooth the class weights via square-root dampening ($\alpha_t \leftarrow \sqrt{\alpha_t}$) to prevent over-correction. The high γ value aggressively down-weights well-classified examples, forcing the model to focus on confusable pairs such as *Pessimistic* vs. *Serious*, thereby optimizing for improved Macro-F1 performance.

5 Results and Analysis

Our official submission achieved a Macro-F1 of 0.1703 on the leaderboard (7th place). The "Gold Dataset" results (Table 3) refer to evaluation on the official test set after labels were released post-submission. On this set, MuRIL achieved a Macro-F1 of 0.2072; the improvement over the leaderboard score is attributed to late-stage refinements in the `difflib` sequence-matching thresholds applied after submission.

5.1 Post-Submission Experiments

The performance of **IndicBERTv2** on the Gold Dataset was Macro-F1: 0.3000 while MuRIL had Macro-F1: 0.2072 and DistilBERT had Macro-F1: 0.1287 for Telugu, which indicates that the knowledge distillation process loses the cross-lingual features important for low-resource style recognition. Based on these results, it can be inferred that the better option would have been to submit IndicBERTv2 instead.

Model	Acc.	Prec.	Rec.	F1
MuRIL (Gold)	0.2093	0.2106	0.2197	0.2072
IndicBERTv2	**0.3100**	**0.3100**	**0.3200**	**0.3000**
mBERT	0.2423	0.2500	0.2600	0.2492
DistilBERT	0.1300	0.1250	0.1350	0.1287
MuRIL (Leaderboard)	*0.1761*	*0.1679*	*0.1865*	*0.1703*

Table 3: Post-submission results on the official test set (rows 1–4) alongside the pre-release leaderboard submission (italicised row).

5.2 Ablation Study

To quantify the contribution of the Turbo Sandwich strategy and the Focal Loss objective, we conducted ablation experiments on the development set using MuRIL. Results are shown in Table 4. Both components independently improve over the baseline, and their combination yields the largest gain. The training and validation loss curves (Figure 2) show that validation loss plateaus after around epoch 6, indicating mild overfitting in later epochs.

5.3 Error Analysis

The confusion matrix in Figure 3 reveals that *Optimistic* and *Inspiring* are relatively well-separated. However, *Authoritative* is frequently misclassified as *Formal*, and *Serious* overlaps with *Pessimistic*.

Example 1 — Authoritative → Formal (misclas-

Configuration	F1	Δ
Plain concat +		
Cross-Entropy (baseline)	0.1421	—
Plain concat +		
Focal Loss	0.1589	+0.0168
Turbo Sandwich +		
Cross-Entropy	0.1672	+0.0251
Turbo Sandwich +		
Focal Loss (submitted)	**0.1894**	**+0.0473**

Table 4: Ablation results on the development set (MuRIL). "Plain concat" concatenates original and modified text with a `[SEP]` separator. F1 = Dev Macro-F1.

Figure 2: Training and validation loss curves for MuRIL over 15 training epochs.

sified):
Modified text: "*meeru ee pani cheyyali*" (You must do this task)
Issue: Both *Authoritative* and *Formal* styles in Telugu employ polite imperative verb endings (e.g., *cheyyali*) and honorific second-person forms. The Turbo Sandwich delta captures the verb form but cannot distinguish the pragmatic force behind it without broader discourse context.

Example 2 — Serious → Pessimistic (misclassified):
Modified text: "*ee samasyalu chala kashtamaina vani anipistundi*" (These problems seem very difficult)
Issue: Negatively valenced content in Telugu often lacks the explicit lexical anchors that differentiate *Serious* concern from *Pessimistic* resignation. The model conflates low-valence vocabulary with the Pessimistic class.

6 Conclusion

This paper compares two multilingual Transformers on prompt style recovery task for the Telugu

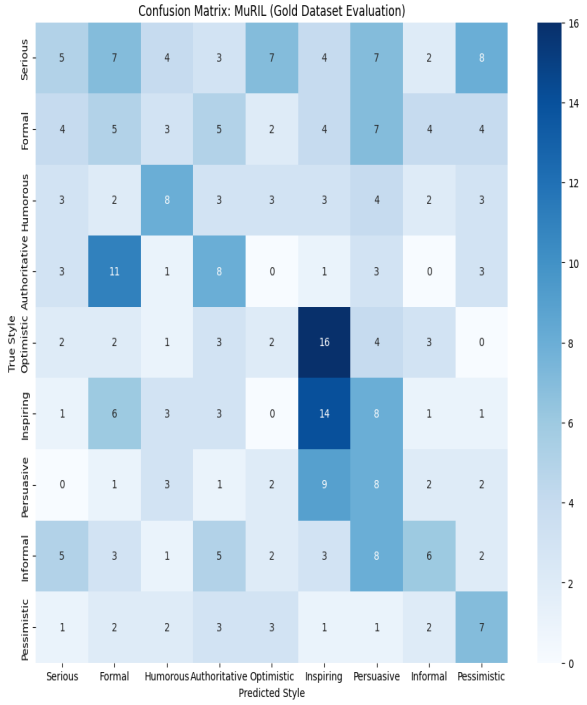


Figure 3: Confusion matrix for MuRIL on the Gold Dataset. Rows are true labels, columns are predicted labels.

language, a low-resource nine-class classification task. We propose the Turbo Sandwich preprocessing strategy that can identify lexical deltas by sequence matching and yields better results than simple concatenation with Focal Loss. While the official submission of MuRIL is done, the results from post-submission experiments with IndicBERTv2 prove that it performs better for this task with a Macro-F1 of 0.3000 than MuRIL with 0.2072 for Macro-F1. The implications of sequence matching threshold ablation, early stopping using validation loss, and extension of Turbo Sandwich approach to other Dravidian languages will be investigated in future work.

Limitations

The small dataset (3,000 training instances) limits the generalizability of our findings to larger or more diverse corpora. Telugu-English code-mixing may additionally disadvantage models with predominantly monolingual Indic pretraining. The confusion matrix in Figure 3 may suffer from low resolution in print; we refer readers to our GitHub repository for a high-resolution version.

Ethics Statement

All data used in this work were provided by the shared task organizers. We did not collect or an-

notate new data. Our models are trained for text classification only and do not generate new content. We are committed to building culturally sensitive NLP systems that represent low-resource language communities equitably.

References

Premjith B, Bharathi Raja Chakravarthi, Prasanna Kumar Kumaresan, Saranya Rajiakodi, Sai Prashanth Karnati, Sai Rishith Reddy Mangamuru, and Chandu Janakiram. 2024. Findings of the shared task on hate and offensive language detection in Telugu codemixed text (HOLD-Telugu)@DravidianLangTech 2024. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 49–55. Association for Computational Linguistics.

Pramit Bhattacharyya and Arnab Bhattacharya. 2025. BanglaByT5: Byte-level modelling for Bangla. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5551–5560. Association for Computational Linguistics.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *Proceedings of the 30th USENIX Security Symposium*, pages 2633–2650. USENIX Association.

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

DravidianLangTech Organizing Committee. 2024. Overview of the shared task on prompting for Telugu at DravidianLangTech 2024. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages (DravidianLangTech 2024)*, pages 1–12. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Pushpak Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. *Findings of EMNLP*.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Mandhale, Ashwin Mantena, Shachi Bhardwaj, Vinay Niyogi, Mirac Sharifi, Chung-Cheng Pan, and Jason Riesa. 2021. Muril: Multilingual representations for indian languages. In *arXiv preprint arXiv:2103.10730*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*.

Faizad Ullah, Safiullah Faizullah, Imdad Ullah Khan, Turki Alghamdi, Toqeer Ali Syed, Ahmad B. Alkhodre, Muhammad Sohaib Ayub, and Asim Karim. 2025. Prompt-based fine-tuning with multilingual transformers for language-independent sentiment analysis. *Scientific Reports*, 15:20834.

Shubham Vatsal, Harsh Dubey, and Aditi Singh. 2025. Multilingual prompt engineering in large language models: A survey across nlp tasks. *arXiv preprint arXiv:2505.11665*.

R. Vemula, M. Nuthi, and M. Srivastava. 2022. TeQuAD: Telugu question answering dataset. In *Proceedings of the 19th International Conference on Natural Language Processing (ICON 2022)*, pages 300–307. Association for Computational Linguistics.

SUPERNOVA@DravidianLangTech 2026: Transformer and Ensemble Approaches for Abusive Tamil Text Detection Targeting Women

Kiruthika K[1], Roahiyaa T[1], Premjith B[1]
[1]Amrita School of Artificial Intelligence,
Amrita Vishwa Vidyapeetham, Coimbatore, India
cb.sc.u4aie24329@cb.students.amrita.edu
cb.sc.u4aie24043@cb.students.amrita.edu
b_premjith@cb.amrita.edu

Abstract

Abusive language targeting women on Tamil social media is a growing concern that necessitates automated detection systems capable of handling low-resource, code-mixed, and morphologically rich text. This paper presents the SUPERNOVA system submitted to the shared task on Abusive Tamil Text Targeting Women on Social Media at DravidianLangTech@ACL 2026. We investigate three complementary approaches: (1) fine-tuning MuRIL with class balancing and label smoothing, (2) MuRIL contextual embeddings combined with XGBoost and decision threshold tuning, and (3) a lightweight ensemble of character-level TF-IDF and SentenceBERT features with Random Forest and Extra Trees. Our best system achieves an accuracy of 0.8007 and a macro F1-score of 0.7994, ranking 11th among all participating teams. These results highlight the effectiveness of multilingual transformer representations combined with ensemble techniques for the detection of abusive text on Tamil social networks. The code is publicly available at https://github.com/Kiruthi001/SuperNova-DravidianLangTech-ACL2026.

1 Introduction

The rapid growth of social media has created new avenues for hate speech and abusive language, particularly targeting women. Tamil, spoken by over 80 million people across India and Sri Lanka, is widely used on platforms such as YouTube, Twitter/X, and Facebook. While this digital expansion has enabled greater communication and visibility, it has also amplified instances of targeted harassment and misogynistic content directed at Tamil-speaking women.

Automatically detecting such abusive content is both essential and challenging. Tamil social media text is linguistically complex due to its agglutinative morphology, frequent code-mixing with English, and the use of both native Tamil script and Roman transliteration. Additionally, the informal and rapidly evolving nature of online language further complicates automatic moderation systems.

The DravidianLangTech@ACL 2026 shared task (Rajiakodi et al., 2026) addresses this issue by providing annotated Tamil social media data for binary classification (*Abusive* vs. *Non-Abusive*) specifically targeting women. To tackle this challenge, we investigate three complementary systems: (1) an end-to-end fine-tuned MuRIL model, (2) a hybrid MuRIL and XGBoost approach, and (3) a CPU-efficient ensemble integrating character-level TF-IDF and SentenceBERT features with classical classifiers. Our best system ranks 11th among participating teams, achieving an accuracy of 0.8007 and a macro F1-score of 0.7994, suggesting that Indic-pretrained transformers combined with ensemble strategies are effective for abusive Tamil text detection in social media settings.

2 Related Work

Abusive language detection in Dravidian languages has been explored across multiple DravidianLangTech shared tasks (Chakravarthi et al., 2020). Early systems relied on classical methods such as TF-IDF with SVM and Random Forests, which showed competitive performance for morphologically rich languages. However, the emergence of pre-trained multilingual transformers significantly advanced the state of the art. mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) demonstrated strong cross-lingual transfer, while MuRIL (Khanuja et al., 2021), trained on 17 Indian languages, consistently outperformed generic models on Indic benchmarks. Furthermore, SentenceBERT (Reimers and Gurevych, 2019) provided efficient semantic embeddings that complement ensemble classifiers in resource-constrained settings. Our work builds on these findings by combining transformer-based representations with ensemble

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 376–380
July 3, 2026 ©2026 Association for Computational Linguistics

strategies tailored for Tamil abusive text detection.

3 Methodology

Our methodological design aims to compare contextual deep learning approaches with classical machine learning models under a unified preprocessing and evaluation framework for abusive Tamil text detection.

3.1 Dataset

The dataset was released as part of the shared task on *Abusive Tamil Text Targeting Women on Social Media* at DravidianLangTech@ACL 2026 (Rajiakodi et al., 2026). It consists of Tamil YouTube comments annotated as *Abusive* or *Non-Abusive* by domain experts following predefined annotation guidelines. The dataset contains native Tamil script, Roman transliterations, and code-mixed Tamil-English content, reflecting the diversity of real-world social media language. Table 1 summarizes the dataset statistics.

Split	Total	Abusive	Non-Abusive
Train	25945	12971	12974
Test	913	441	472

Table 1: Dataset statistics.

3.2 System Architecture

Figure 1 illustrates the overall pipeline shared across all three runs. The raw Tamil text first passes through a common preprocessing stage. The cleaned text is then fed into three independent modeling pipelines, each representing a distinct approach. All three systems are evaluated using the same classification metrics defined by the shared task organizers. Character-level TF-IDF is particularly useful for Tamil due to its rich morphology and frequent spelling variations in informal social media text.

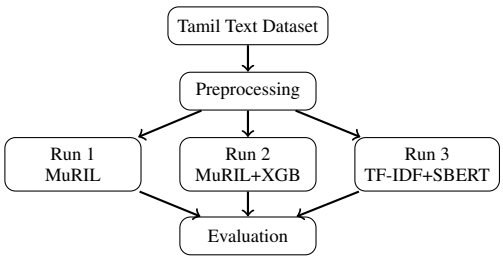


Figure 1: Overall system architecture.

3.3 Data Preprocessing

All input text undergoes the following steps: (1) lowercase normalization; (2) removal of URLs, @mentions, and #hashtags; (3) retention of Tamil Unicode characters (U+0B80–U+0BFF) and alphanumeric characters; and (4) collapsing of extra whitespace.

For transformer-based models, text is tokenized using subword tokenization with padding or truncation to a fixed maximum sequence length. For classical machine learning models, cleaned text is transformed into numerical TF-IDF feature vectors.

3.4 Run 1: MuRIL Fine-Tuning

We fine-tune `google/muril-base-cased` (Khanuja et al., 2021) for binary sequence classification. MuRIL is pre-trained on 17 Indian languages including Tamil, making it well-suited for this task. To address class imbalance, the abusive class is oversampled by duplication.

Training uses an 85/15 stratified split with the AdamW optimizer ($lr = 3 \times 10^{-5}$, weight decay $= 0.01$), cosine learning rate scheduling, label smoothing of 0.1, batch size of 16, and 6 training epochs with FP16 mixed precision. The maximum token length is set to 128.

This end-to-end fine-tuning strategy enables the model to adapt contextual representations specifically to abusive Tamil language patterns and context-dependent expressions.

3.5 Run 2: MuRIL Embeddings + XGBoost

MuRIL is first fine-tuned for 3 epochs using AdamW ($lr = 2 \times 10^{-5}$, maximum length $= 192$). Sentence-level embeddings are then extracted by concatenating the normalized `[CLS]` token representation and the mean-pooled last hidden state, resulting in a 1536-dimensional feature vector.

These embeddings are used to train an XGBoost classifier (Chen and Guestrin, 2016) with 4000 estimators, maximum depth of 4, and learning rate of 0.03. The classification threshold is tuned on the validation set to maximize macro F1-score.

This approach separates representation learning from classification, allowing tree-based boosting methods to operate on strong contextual embeddings without full end-to-end adaptation.

3.6 Run 3: TF-IDF + SentenceBERT + Ensemble

This configuration is designed to operate without GPU resources. It combines two complementary

feature types:

(a) Character-level TF-IDF (n-grams 3–5, maximum 50k features, min_df=5), which captures morphological patterns and is robust to transliteration noise.

(b) Multilingual SentenceBERT embeddings (`paraphrase-multilingual-MiniLM-L12-v2`, 384-dimensional) (Reimers and Gurevych, 2019), which provide cross-lingual semantic representations.

Both feature matrices are horizontally concatenated and passed to a soft-voting ensemble of Random Forest and Extra Trees classifiers (400 estimators each, class weights {0:1.0, 1:1.5}).

This configuration prioritizes computational efficiency and deployability in low-resource settings while maintaining reasonable classification performance.

3.7 Evaluation Metrics

Model performance is evaluated using Accuracy, Precision, Recall, and Macro F1-score in accordance with the shared task guidelines.

Accuracy measures overall correctness but may be misleading in the presence of class imbalance. Therefore, Macro F1-score is considered the primary evaluation metric, as it computes the F1-score independently for each class and then averages them, ensuring equal importance to both the *Abusive* and *Non-Abusive* categories.

Precision reflects how many predicted abusive instances are actually abusive, while Recall indicates how many abusive instances were successfully detected. The F1-score balances both metrics and provides a more comprehensive evaluation of model performance.

4 Experiments

4.1 System Configuration

Run 1 and Run 2 were conducted on Google Colab using an NVIDIA T4 GPU. Run 3 was executed on a CPU-only machine. All experiments use Python 3.10 with PyTorch, HuggingFace Transformers, Scikit-learn, and XGBoost libraries. The code is publicly available at `https://github.com/Kiruthi001/SuperNova-DravidianLangTech-ACL2026`.

4.2 Results

Table 2 presents the validation performance of all three runs on the training data. Table 3 presents the test set results for all three runs. Our Run 1 submission was officially ranked **11th** on the shared task leaderboard.

Run	System	Acc	P	R	F1
Run 1	MuRIL Fine-tune	0.8526	0.8383	0.8309	0.8283
Run 2	MuRIL+XGBoost	0.8222	0.82	0.82	0.8216
Run 3	TF-IDF+SBERT+ET	0.8085	0.81	0.81	0.8100

Table 2: Validation results on training data.

Run	System	Acc	P	R	F1
Run 1	MuRIL Fine-tune	0.8007	0.8031	0.7989	0.7994
Run 2	MuRIL+XGBoost	0.8007	0.8031	0.7989	0.7994
Run 3	TF-IDF+SBERT+ET	0.7415	0.7447	0.7432	0.7413

Table 3: Test set results. Run 1 ranked 11th officially.

4.3 Analysis

The validation results indicate that the transformer-based Run 1 model achieves the highest performance among all three approaches. With a validation macro F1-score of 0.8283, MuRIL fine-tuning demonstrates strong capability in modeling contextual dependencies in Tamil social media text. The use of cosine learning rate scheduling and label smoothing contributes to improved generalization during training.

However, a noticeable performance drop is observed when evaluating on the official test set (from 0.8283 to 0.7994 macro F1). This gap suggests potential domain shift between the training and test distributions or mild overfitting during fine-tuning. Despite this, Run 1 remains the strongest submission and achieves 11th rank on the leaderboard.

Run 2 (MuRIL + XGBoost) achieves slightly lower validation performance compared to Run 1. Although it leverages the same contextual representations, separating feature extraction and classification introduces additional optimization constraints. Unlike end-to-end fine-tuning, XGBoost does not update the transformer parameters during inference, which may limit adaptation to subtle abusive patterns.

Interestingly, Run 2 achieves identical official test scores to Run 1 in the leaderboard results. This suggests that tree-based classification over strong contextual embeddings can approximate end-to-end performance when decision thresholds are carefully tuned.

Interestingly, Run 2 achieves identical official test scores to Run 1 in the leaderboard results.

While Run 1 performs end-to-end fine-tuning, Run 2 relies on fixed MuRIL embeddings followed by an XGBoost classifier. This similarity suggests that the pretrained MuRIL representations are sufficiently expressive, allowing tree-based models to learn effective decision boundaries without full model adaptation. Additionally, the relatively small size of the test set and threshold tuning may contribute to the similar observed performance.

Run 3 demonstrates that classical machine learning approaches remain viable, especially under computational constraints. Character-level TF-IDF effectively captures Tamil morphological structures and spelling variations, including transliterated forms. The addition of SentenceBERT embeddings improves semantic coverage. However, the absence of deep contextual modeling limits its ability to detect implicit abuse, sarcasm, or context- dependent misogyny, resulting in lower macro F1-score (0.7413).

Overall, the results confirm that Indic-pretrained transformer models provide strong inductive bias for Tamil abusive text detection. While ensemble-based classical approaches offer computational efficiency, end-to-end contextual fine-tuning yields the most robust and consistent performance.

5 Conclusion

This paper presented three complementary systems for abusive Tamil text detection targeting women in the DravidianLangTech@ACL 2026 shared task. We compared transformer-based fine-tuning, hybrid embedding-based gradient boosting, and classical ensemble approaches under a unified evaluation framework.

Our best-performing system, based on end-to-end MuRIL fine-tuning, achieved an accuracy of 0.8007 and a macro F1-score of 0.7994, securing 11th rank on the official leaderboard. The results demonstrate that Indic-pretrained transformer models provide strong contextual representations for modeling code-mixed and morphologically rich Tamil social media text.

Additionally, our CPU-efficient ensemble model highlights that competitive performance can be achieved without GPU resources, making it suitable for practical moderation systems in low-resource environments.

Future work will explore data augmentation, context-aware modeling, and larger Indic language models to further enhance abusive text detection performance.

Limitations

Despite achieving competitive performance, our models have several limitations. First, Tamil social media language evolves rapidly, including the use of new slang, emojis, and creative spellings, which may not be adequately captured by models trained on static datasets. Second, transformer-based approaches require GPU resources for efficient training, which may limit reproducibility in low-resource environments. Additionally, we did not explore advanced data augmentation techniques such as back-translation or cross-lingual transfer learning, which could potentially improve generalization performance. Finally, our models operate at the comment level and do not incorporate conversation context, which may be necessary for detecting implicit or context-dependent abuse.

Ethical considerations

This work focuses on detecting abusive language targeting women in Tamil social media. Automated moderation systems must be used carefully to avoid bias or incorrect labeling of benign content as abusive. The models developed in this study are intended to assist human moderators rather than replace them. We use only the anonymized dataset provided by the shared task organizers and emphasize responsible deployment that respects user privacy and fairness.

References

Bharathi Raja Chakravarthi and 1 others. 2020. Corpus creation for sentiment analysis in code-mixed Tamil-English text. In *Proceedings of SLTU-CCURL 2020*.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of KDD 2016*.

Alexis Conneau and 1 others. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL 2020*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL 2019*.

Simran Khanuja and 1 others. 2021. MuRIL: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Saranya Rajiakodi, Shunmuga Priya Muthusamy Chinnan, Rajalakshmi R., Kathiravan Pannerselvam, Bhuvaneswari Sivagnanam, Jananayagan V, Charmathi Rajkumar, R Ramesh Kannan, and Bharathi Raja Chakravarthi. 2026. From Comments to Harm: A Findings Report on Abusive Tamil Text Targeting Women on Social Media Shared Task- DravidianLangTech@ACL 2026. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of EMNLP 2019*.

SYNAPSE@DravidianLangTech 2026: Multi-Level Political Meme Classification for Tamil and Malayalam

Suriya K P[1], Durai Singh K[1], Gnanasabesan G[1], Ganesh Sundhar[1], Hari Krishnan N[1], Jyothish Lal G[1]

[1]Amrita School of Artificial Intelligence, Coimbatore, Amrita Vishwa Vidyapeetham, India

{cb.en.u4aie22164, cb.en.u4aie22167, cb.en.u4aie22018, cb.en.u4aie22017 cb.en.u4aie22020}@cb.students.amrita.edu, g_jyothishlal@cb.amrita.edu

Abstract

Political memes in Tamil and Malayalam present unique multimodal challenges for automated understanding, combining visual context with code-mixed, culturally grounded text. We present **SYNAPSE**, our system for the DravidianLangTech@ACL 2026 shared task on multi-level political meme classification. The task requires hierarchical classification of memes along two levels: Level 1 identifies the political stance (Support/Praise vs. Troll/Oppose), and Level 2 identifies the target (individual person vs. party). Our approach fine-tunes the **Qwen3-VL-2B-Instruct** vision-language model using parameter-efficient LoRA adapters on task-specific multimodal data, with structured output prompting for hierarchical label prediction. We report results for both Tamil and Malayalam subtracks. For Malayalam, our system achieves a Level 1 F1 of 0.9200 and Level 2 F1 of 0.4256 (Avg-F1: 0.6728, Rank 5). For Tamil, our system achieves a Level 1 F1 of 0.7840 and Level 2 F1 of 0.4885 (Avg-F1: 0.6362, Rank 14).

Keywords: Political meme classification, multimodal NLP, vision-language models, Tamil, Malayalam, LoRA fine-tuning, DravidianLangTech.

1 Introduction

Political memes have become a potent instrument in online political discourse, particularly in multilingual and low-resource languages such as in Tamil and Malayalam social media. Unlike plain text, memes combine image and text in culturally situated ways that make their automated interpretation more challenging (Kiela et al., 2020). In the Tamil and Malayalam social media landscape, political memes routinely employ code-switching, sarcasm, regional symbols, and implicit references to political events that are vague to standard NLP systems (Francis et al., 2025).

The DravidianLangTech@ACL 2026 shared task on Multi-Level Political Meme Classification addresses this gap by making it a hierarchical classification challenge over Tamil and Malayalam political memes. Each meme must be classified along two levels simultaneously: Level 1 states whether the meme expresses *Support/Praise* or *Troll/Oppose* toward some political entity or person, and Level 2 identifies the target of that stance, specifically whether it is directed at an individual person or a party. The fine-grained nature of Level 2, with categories including *Troll Against Person*, *Troll Against Party*, *Support for Person*, and *Support for Party*, requires systems to go beyond surface level sentiment understading toward in detail political target identification.

We present SYNAPSE, a multimodal classification system based on parameter-efficient fine-tuning of the Qwen3-VL-2B-Instruct vision-language model (Qwen Team, 2025). Our system formats hierarchical classification as structured generation: a single pass of the model produces both Level 1 and Level 2 predictions via a constrained output template. This approach avoids the need for decoupled classification heads or two-stage inference pipelines, making it computationally efficient and straightforward to deploy.

Our key contributions are as follows:

1. **Multimodal fine-tuning** of a compact vision-language model (Qwen3-VL-2B) using LoRA adapters, enabling effective learning from small task-specific datasets without full model retraining.

2. **Structured output prompting** that defines joint Level 1 and Level 2 predictions in a single generation pass via a fixed template format, ensuring consistent hierarchical label extraction.

3. **Empirical comparison** of multiple ap-

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 381–386
July 3, 2026 ©2026 Association for Computational Linguistics

proaches including knowledge distillation from Qwen-8B and chain-of-thought prompting with translation, demonstrating that direct LoRA fine-tuning on the 2B model outperforms these alternatives on this task.

2 Related Work

2.1 Multimodal Meme Understanding

Meme understanding has received increasing attention in the NLP and computer vision research. Kiela et al. (2020) introduced the Hateful Memes dataset, establishing multimodal classification of meme content into a benchmark challenge and prooving that vision-language approach outperforms single modal baselines. Subsequent work has explored late fusion, attention mechanisms, and pre-trained vision-language models for meme understanding tasks (Pramanick et al., 2021).

2.2 Political Sentiment in Dravidian Languages

Sentiment analysis in Dravidian languages has been a recurring theme in the DravidianLangTech workshop series (Chakravarthi et al., 2022). Prior work has used transformer-based models including mBERT, IndicBERT, and XLM-RoBERTa for code-mixed Tamil and Malayalam text (Kakwani et al., 2020; Conneau et al., 2020). However, most prior systems treat the problem as purely text-based problem, neglecting the visual component of meme content.

2.3 Vision-Language Models for Low-Resource languages

Large-scale vision-language models (VLMs) such as LLaVA (Liu et al., 2023), InstructBLIP (Dai et al., 2023), (Kumar et al., 2024), (Premjith et al., 2023), and the Qwen-VL series (Bai et al., 2023) have demonstrated strong performance on image-text tasks with appropriate fine-tuning. Parameter-efficient fine-tuning methods such as LoRA (Hu et al., 2022) make it feasible to adapt large models to low-resource multimodal tasks without prohibitive computational cost. Our approach builds on this paradigm, applying LoRA fine-tuning to Qwen3-VL-2B-Instruct for politically in depth, Dravidian-language meme classification.

3 Task and Dataset Description

3.1 Task Definition

The shared task requires hierarchical classification of political memes along two levels:

- **Level 1:** Binary stance classification — *Support/Praise* vs. *Troll/Oppose*.
- **Level 2:** Target identification — whether the stance is directed at an *Individual Person* or a *Party* (or *Intersection* in the Malayalam track).

The full Level 2 label taxonomy across the two tracks is as follows. For Tamil: *Troll/Oppose Against Person*, *Troll/Oppose Against Party*, *Support for Person*, and *Support for Party*. For Malayalam: the corresponding categories additionally include *Intersection*, referring to memes that target an intersection of a person and a party simultaneously.

Evaluation uses macro-averaged precision, recall, and F1 score, computed independently for Level 1 and Level 2 predictions, and reported as per the scikit-learn classification report. The final ranking is based on the average of Level 1 and Level 2 macro F1 scores.

3.2 Datasets

Two separate datasets are provided for Tamil and Malayalam. Each meme is represented by an image file and a structured label file (Rajiakodi et al., 2026). Table 1 summarises partition sizes across both language tracks.

Language	Split	Instances	L2 Classes
Tamil	Train	803	4
	Test	201	4
Malayalam	Train	500	5
	Test	100	5

Table 1: Dataset statistics for Tamil and Malayalam tracks.

3.3 Class Distribution

Both datasets exhibit notable class imbalance. In the Tamil training set, *Troll/Oppose Against Person* constitutes the majority class (547 instances), followed by *Troll/Oppose Against Party* (146), *Support for Person* (86), and *Support for Party* (24). In the Malayalam training set, *Against Individual Person* is also dominant (315), while *Support for Party* and *Support for Individual Person* have far fewer

instances (10 and 12 respectively). This imbalance motivates the use of macro-averaged metrics and poses a significant challenge for minority-class performance at Level 2.

4 Methodology

Our system fine-tunes the Qwen3-VL-2B-Instruct vision-language model (Qwen Team, 2025) using Low-Rank Adaptation (LoRA) (Hu et al., 2022) to perform joint hierarchical classification across both levels in a single structured generation pass.

4.1 Base Model

We use **Qwen3-VL-2B-Instruct** as our backbone. This model provides a compact yet capable multimodal encoder-decoder architecture, jointly processing image and text inputs via a visual encoder and a transformer decoder. Its instruction-tuned variant supports structured dialogue-format inputs, making it well-suited for template-based output generation. Images are processed at a maximum resolution of 512×512 pixels to balance visual fidelity with computational efficiency.

4.2 Data Preprocessing

Training labels are normalised to a unified vocabulary prior to fine-tuning. Level 1 labels are mapped to `SUPPORT` and `CRITICISE`, while Level 2 labels are mapped to `PERSON`, `PARTY` and `BOTH`. Each training instance is converted into a structured conversation consisting of a user turn containing the meme image and a fixed instruction prompt, and an assistant turn containing the ground-truth structured output:

```
INTENT: [SUPPORT|CRITICISE]
TARGET: [PERSON|PARTY|BOTH]
```

This format enforces a predictable output structure that can be deterministically parsed at inference time, eliminating the ambiguity of free-form generation.

4.3 LoRA Fine-Tuning

We apply LoRA to the query and value projection matrices (`q_proj` and `v_proj`) of the transformer decoder. Fine-tuning hyperparameters are as follows: rank $r = 4$, $\alpha = 8$, dropout $= 0.15$. Training uses a per-device batch size of 1 with gradient accumulation over 8 steps (effective batch size 8), a learning rate of 5×10^{-5}, and a weight decay of 0.1. Models are trained for 500 steps with bfloat16 mixed precision. Only the LoRA adapter weights are saved after training, keeping the checkpoint lightweight and the base model frozen.

4.4 Inference and Output Parsing

At inference time, each test meme is passed through the model with the same instruction prompt used during fine-tuning. Generation is performed greedily (`do_sample=False`) with a maximum of 150 new tokens. The output is parsed line-by-line: the `INTENT:` line yields the Level 1 prediction and the `TARGET:` line yields the Level 2 prediction. Unknown or malformed outputs are assigned a fallback label of `UNKNOWN`.

4.5 Explored Alternatives

We investigated two additional approaches that were ultimately outperformed by direct LoRA fine-tuning:

- **Knowledge distillation from Qwen3-VL-8B:** We attempted to use the 8B variant as a teacher model to generate soft labels or explanations for training the 2B student. While the 8B model produced higher-quality zero-shot outputs, the distillation process did not yield consistent improvements over direct task fine-tuning of the 2B model, likely due to domain mismatch and the small training set size.

- **Chain-of-thought prompting with translation:** We explored prompting the model to first translate the Tamil or Malayalam meme text to English and then reason step-by-step before predicting the final labels. This approach consistently underperformed, as translation introduced errors on code-mixed and colloquial inputs, and the additional reasoning steps were not reliably grounded in the meme's visual content.

These comparisons confirm that direct, targeted fine-tuning on the task data with structured output constraints is the most effective strategy for this low-resource multimodal classification problem.

5 Results and Analysis

5.1 Evaluation Metric

Performance is reported using macro-averaged precision (P), recall (R), and F1 score separately for Level 1 and Level 2 predictions. Accuracy (ACC) is also reported. The final ranking metric is the

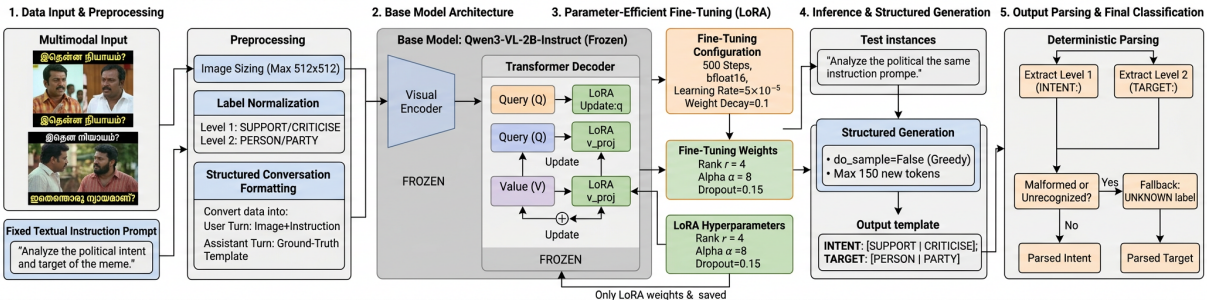


Figure 1: Overview of the proposed SYNAPSE architecture for multi-level political meme classification. The pipeline processes each meme through parallel image and text streams, feeds them into a Qwen3-VL-2B-Instruct vision-language model fine-tuned with LoRA adapters on query/value projections (base model frozen), and produces hierarchical predictions via structured output prompting: Level 1 stance (Support vs. Troll) and Level 2 target (Person vs. Party).

average of Level 1 and Level 2 macro F1 scores:

$$\text{Avg-F1} = \frac{\text{F1}_{\text{Level1}} + \text{F1}_{\text{Level2}}}{2} \quad (1)$$

5.2 Official Results

Tables 2 and 3 report our official shared task results alongside the top-ranked systems for the Malayalam and Tamil subtracks respectively.

Team	L1-F1	L2-F1	Avg-F1	Rank
SYNAPSE	**0.9200**	**0.4256**	**0.6728**	**5**

Table 2: Official results - Malayalam subtrack (selected teams).

Team	L1-F1	L2-F1	Avg-F1	Rank
SYNAPSE	**0.7840**	**0.4885**	**0.6362**	**14**

Table 3: Official results - Tamil subtrack (selected teams).

5.3 Analysis

Across both subtracks, Level 1 performance substantially exceeds Level 2 performance. This is expected: binary stance classification (Support vs. Troll) is a simpler task with higher inter-class separability, whereas Level 2 target identification requires resolving fine-grained distinctions between person- and party-directed stances that often co-occur in the same meme.

The Malayalam subtrack achieved a higher Level 1 F1 (0.9200) than Tamil (0.7840), possibly reflecting a cleaner visual signal in the Malayalam memes or a more balanced training distribution at Level 1. The Tamil Level 2 F1 (0.4885), however, slightly exceeds the Malayalam Level 2 F1 (0.4256), suggesting that the Tamil training data provides somewhat stronger supervision for fine-grained target identification, despite its severe class imbalance at the minority categories.

The low Level 2 scores across all submitted systems reflect the inherent difficulty of this sub-problem: distinguishing person-directed from party-directed political stance in memes requires the system to recognize political figures and party symbols from visual content, a capability that compact VLMs with limited fine-tuning data may struggle to acquire reliably. Future work could address this by incorporating entity recognition or political knowledge grounding.

6 Conclusion

We presented SYNAPSE, a LoRA-fine-tuned vision-language system for multi-level political meme classification in Tamil and Malayalam. By framing hierarchical classification as structured text generation from a multimodal input, our system avoids the complexity of multi-head architectures while achieving competitive performance. Our system ranked 5th on the Malayalam subtrack (Avg-F1: 0.6728) and 14th on the Tamil subtrack (Avg-F1: 0.6362).

Comparative experiments showed that direct LoRA fine-tuning on task data outperforms knowledge distillation from a larger model and chain-of-thought prompting with translation, highlighting the importance of task-specific adaptation over general reasoning strategies for this low-resource multimodal domain.

Code: https://github.com/SURIYA-KP/Poli-Meme

Limitations

1. **Small training sets:** With 803 Tamil and 500 Malayalam training instances, both datasets are small for multimodal fine-tuning. Minority classes at Level 2 have very few examples, directly limiting classification performance on those categories (Francis et al., 2026).

2. **Label space inconsistency across languages:** The Malayalam dataset includes an additional *Intersection* class at Level 2, but the Tamil dataset doesn't contain this class. Due to this mismatch in label taxonomies, we had to train separate models for each language, for the best performance instead of a unified model over the combined dataset, limiting its generalization.

3. **Visual grounding of political entities:** The system has no explicit mechanism to identify which political figure or party appears in a meme's image. Level 2 predictions rely on the model's implicit visual knowledge acquired during pretraining, which may be inconsistent for regional political figures less represented in the pretraining corpus.

4. **Resolution constraint:** Capping image resolution at 512×512 pixels may discard fine-grained textual content in memes where embedded text is small or dense, which is common in Tamil political memes.

Acknowledgments

We thank the organizers of DravidianLangTech@ACL 2026 for curating the Tamil and Malayalam political meme datasets and hosting the shared task. We also gratefully acknowledge the computational resources provided by our institution.

References

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Sajeetha Thavareesan, Dhivya Chinnappa, Durairaj Thenmozhi, Elizabeth Sherly, and 1 others. 2022. Overview of the shared task on sentiment analysis for dravidian languages in code-mixed text. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 1–9.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*, pages 8440–8451.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36.

Meclin A. Francis, Ayswarya R. Kurup, B. Premjith, and Bharathi Raja Chakravarthi. 2025. Multimodal fake news classification in tamil using fact-checked social media content and cost-sensitive learning. *IEEE Access*, 13:157477–157510.

Meclin A. Francis, Ayswarya R. Kurup, B. Premjith, Bharathi Raja Chakravarthi, and Saranya Rajiakodi. 2026. Tamilfacts: A comprehensive multimodal dataset of fact-checked social media content in tamil language. In *Speech and Language Technologies for Low-Resource Languages*, pages 167–182. Springer.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *Proceedings of ICLR*.

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*, pages 4948–4961.

Douwe Kiela, Hamed Firooz, Aravind Natesan Rau, Virginie Testuggine, Alyssa Mensch, Devvret Mahajan, Josh Meyer, Maximilian Nickel, Dhruv Mahajan, and Laurens van der Maaten. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Advances in Neural Information Processing Systems*, volume 33, pages 2611–2624.

MR Dinesh Kumar, Pillalamarri Akshaya, R Saivarsha, NT Shrish Surya, B Premjith, V Sowmya, and G Jyothish Lal. 2024. Enhanced vision language model for visual question answering in medical images. In *International Conference on Data Science and Applications*, pages 225–236. Springer.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36.

Shraman Pramanick, Shivam Sharma, Dimitar Dimitrov, Md. Shad Akhtar, Preslav Nakov, and Tanmoy

Chakraborty. 2021. Momenta: A multimodal framework for detecting harmful memes and their targets. In *Findings of EMNLP*, pages 4439–4455.

B Premjith, G Jyothish Lal, V Sowmya, Bharathi Raja Chakravarthi, Rajeswari Natarajan, K Nandhini, Abirami Murugappan, Bharathi B, M Kaushik, Prasanth Sn, Aswin Raj R, and Vijai Simmon S. 2023. Findings of the shared task on multimodal abusive language detection and sentiment analysis in tamil and malayalam. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, pages 72–79.

Qwen Team. 2025. Qwen3-vl technical report. *arXiv e-prints*.

Saranya Rajiakodi, Shunmuga Priya Muthusamy Chinnan, Premjith B, Subalalitha CN, Rahul Ponnusamy, Anshid K A, Bhuvaneshwari Sivagnanam, Jananayagan V, Bharathi Raja Chakravarthi, Ragavan N, and Santhini P. 2026. Overview of the shared task on multilevel political meme classification in tamil and malayalam. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

TamilEcho_Political@DravidianLangTech 2026: Hybrid XLM-RoBERTa with Sarcasm-Aware Feature Fusion for Political Multiclass Sentiment Analysis in Tamil X

Kanimozhi Selvi C S[1], **Inigashree N S**[1], **Moneissh A G**[1], **Kavinraj J**[1]
[1]Department of Computer Science and Engineering
Kongu Engineering College, Perundurai, India
kanimozhi@kongu.ac.in
inigashreens.22cse@kongu.edu
moneisshag.22cse@kongu.edu
kavinrajj.22cse@kongu.edu

Abstract

Political sentiment analysis of Tamil tweets is challenging due to the informal use of language, sarcasm, emoji-driven sentiment shifts, and a high degree of class imbalance. This paper introduces our system for the Shared Task on Political Multiclass Sentiment Analysis of Tamil X (Twitter) Comments at DravidianLangTech@ACL 2026. We employ a hybrid design comprising contextual representations from XLM-RoBERTa in combination with lexical TF-IDF features and sarcasm-aware emoji indicators. We also use domain-aware hashtag expansion to provide additional political context and apply class-weighted label smoothing to address data imbalance. On the official test set, our system achieves a Macro-F1 score of 0.3559, ranking 10th among participating teams. The findings indicate that combining semantic, lexical, and pragmatic features enhances fine-grained classification of political sentiments in Tamil.

Keywords: Political Comments, Tamil Tweets, Natural Language Processing, XLM-RoBERTa, TF-IDF, Emoji Features, Sentiment Analysis.

1 Introduction

Social media platforms such as X (Twitter) have taken center stage in political discourse, where users are able to share their views and communicate with political actors in real time. Voter attitudes and campaign plans have been analyzed using automated political sentiment analysis (Tumasjan et al., 2010). The effectiveness of social media analysis in terms of modeling political discourse patterns has also been proven with the help of computational methods (Stieglitz and Dang-Xuan, 2012).

Tamil is a Dravidian language which is mainly spoken in Tamil Nadu, India and some in Sri Lanka, Singapore and Malaysia. Despite the abundant literary tradition Tamil is relatively under-reserved in terms of annotated corpora and pre-trained language models. Tweets about political activities usually have spelling differences, code-mixing in English, sarcasm, emojis, and colloquialism that greatly complicate classification. Emerging research on Tamil and other languages in India has emphasized the effect of code-mixing and informal text written in social media in sentiment classification effectiveness (Sreelakshmi et al., 2024). These properties necessitate models that can be used to represent contextual semantics as well as pragmatic signs.

The shared task on Political Multiclass Sentiment Analysis of Tamil X (Twitter) Comments was held toward this purpose as a part of DravidianLangTech@ACL 2026 (Vegupatti et al., 2026). The task includes the categorization of tweets into seven sentiment classes and the performance of systems is evaluated based on Macro-F1 because there is class imbalance. We suggest a hybrid model in this work that combines contextual embedding with transformers, lexical and emoji-based features to contextualize the Tamil political discourse.

2 Related Work

The study of political sentiment analysis has been a major research area in recent years especially in the identification and comprehension of the public opinion and biasness in social media. Prathvi et al. (Prathvi et al., 2024) reported that the Macro-F1 score of ensemble models, which were run using TF-IDF with n-grams, was 0.260 and 0.550, respectively, on code-mixed Tamil and Tulu sentiment analysis. The correctly extraction of lexical features methods in low-resource and code-mixed environment was proven. Kumar et al. (Kumar et al., 2017) developed a sentiment analysis BiLSTM-CNN model to analyze Malayalam tweets and reached an accuracy of 0.9824, which has made it possible to adopt deep neural archi-

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 387–391
July 3, 2026 ©2026 Association for Computational Linguistics

tecture on Dravidian languages. Chakravarthi et al. (Chakravarthi et al., 2021) obtained a weighted F1 score of 0.711 in Tamil-English sentiment analysis, which indicates the problems of multilingual and informal political speech. Rajalakshmi et al. (Rajalakshmi et al., 2022) demonstrated that MuRIL through emoji based sentiment representations showed that explicit emoji model increases code-mixed Tamil text classification. Finally, Angdresey et al. (Angdresey et al., 2025) proposed a hybrid approach incorporating tagging models with BERT representations, random oversampling, and Multinomial Naïve Bayes, achieving an accuracy of 85.155 % and AUC score of 96.80 on political sentiment analysis of YouTube comments.

3 Dataset

The dataset given by the organizers of the shared task entails 5,440 Tamil political tweets, which were gathered in X (Twitter). All tweets are coded into one of seven categories of sentiments, namely Substantiated, Sarcastic, Opinionated, Positive, Negative, Neutral, and None of the Above. The data is divided into the train (4,352 tweets, 80%), development (544 tweets, 10%), and test (544 tweets, 10%) groups. The large percentage of the dataset is constituted by Opinionated and Sarcastic tweets, with the Substantiated and None of the Above classes being relatively low. The imbalance between classes in the training data is depicted in Figure 1. This tradeoff encourages the application of Macro-F1 to be the evaluation metric.

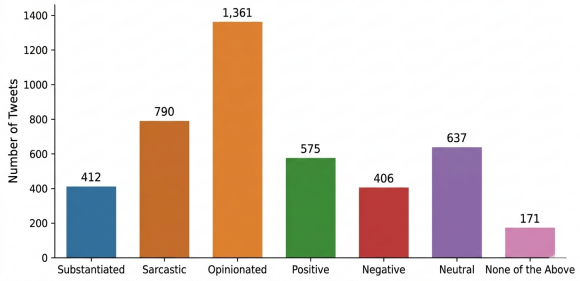


Figure 1: Class distribution in the training dataset.

4 Methodology

4.1 Pre-processing

Tamil political tweets are informal and they are also full of spelling variants, short forms, emojis, and politically charged hashtags. These attributes introduce noise and hidden contextual cues that may adversely influence downstream sentiment classification. So, a specific preprocessing approach is used to enhance the readability of the text and background knowledge. The general process of the preprocessing and feature engineering is shown in Figure 2.

Hashtag Knowledge Expansion:

Political hashtags can also have implicit contextual information, not explicitly found in the text of the tweet. An example would be the use of the hashtag #DMK, #BJP,#NTK and so on to refer to a particular political party or leader. In our method, we augment regularly reoccurring political hashtags with a manually defined dictionary that is written in code. As an example, the #DMK can be mapped to Dravida Munnetra Kazhagam ruling party Tamil Nadu. This expansion which is rule-based assists in including more semantic context and enhances contextual representation in transformer-based models.

Identification of Sarcasm with the help of emojis:

Emojis is important in conveying sarcasm, mockery, and emotional tone in the Tamil politics. Implicit sentiment cues are commonly done with emojis like laughing, angry, folded-hands emojis, and thinking emojis. In order to express this phenomenon explicitly, we derive two emoji-related features: the number of emojis in the tweet in total and the number of emojis denoting sarcasm. These features are pragmatic sentiment predictors and assist the classifier to differentiate among sarcastic and literal sentiment.

4.2 Feature Extraction

Once preprocessed, every tweet is coded into various numerical descriptions of complementary linguistic characteristics. We choose a hybrid method of representation combining transformer embeddings, TF-IDF features, and emoji-based features as shown in Figure 3.

TF-IDF Representation of features:

The importance of lexical meaning in comparison to the corpus is captured with the help of Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF puts more emphasis on words that occur more in a document than in the entire dataset, and therefore focus on discriminative political words. TF-IDF is applied to identify lexical significance. In the case of semantic representation, we use

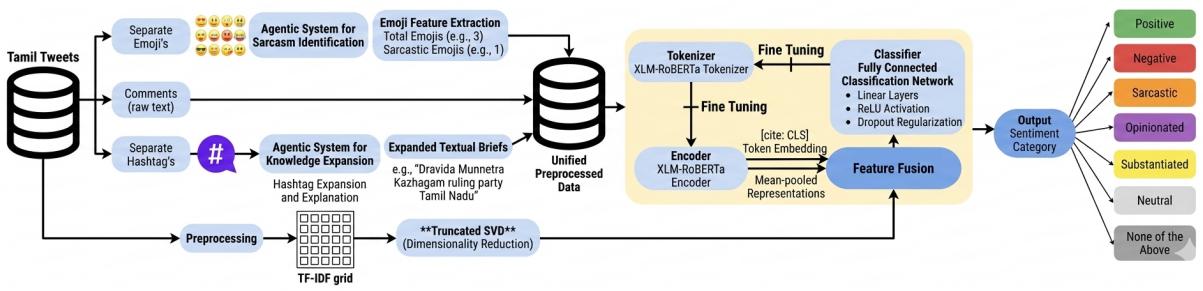


Figure 2: Proposed workflow for political sentiment analysis in Tamil X (Twitter) comments.

XLM-RoBERTa. TF-IDF of a term t in document d is calculated as:

$$TF{-}IDF(t,d) = TF(t,d) \times \log\left(\frac{N}{DF(t)}\right) \quad (1)$$

TF(t,d) is the frequency of a term in a document d, N is the number of documents and DF(t) is the number of documents that contain term t. TF-IDF especially is useful in extracting domain-specific political expressions that the transformer models might not necessarily highlight.

SVD as a form of dimensionality Reduction:

As TF-IDF vectors are both high-dimensional and sparse, Truncated Singular Value Decomposition (SVD) is used to trim the dimensionality without compromising the most informative ones. This step increases noise reduction, computation efficiency and generates a small dense form that can be utilized by neural feature fusion.

Contextual Embeddings based on XLM-RoBERTa:

To achieve semantic presentation, we use XLM-RoBERTa, which is a multilingual transformer that has been trained on multilingual cross-linguistic corpora. Two complementary contextual representations are obtained, specifically that (i) token embedding [CLS], which is a representation of global sentence-level semantics, and (ii) mean-pooled representational embeddings of all tokens, which represent distributed contextual representations. These representations enable the model to exhibit holistic and fine-grained cues in the form of semantics in the tweet.

Feature Design Justification: We combine XLM-RoBERTa (semantic), TF-IDF (lexical), and emoji features (pragmatic) as they provide complementary information, which is more robust to noisy social media text.

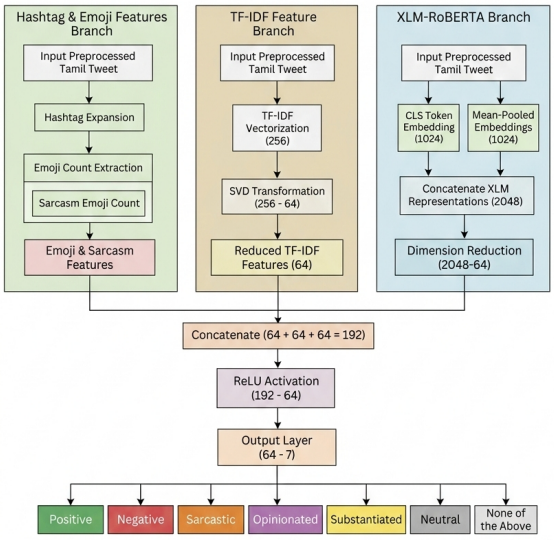


Figure 3: Architecture of the hybrid XLM-RoBERTa-based classification model.

5 Model and Training

We suggest a hybrid architecture to combine contextual embeddings in both XLM-RoBERTa and TF-IDF and emoji-based sarcasm. This architecture allows this design to obtain semantic information of transformer-based representations as well as the lexical significance and the pragmatic cues. The general system design is depicted in Figure 3. The system is comprised of a transformer-based encoder and statistical feature module. The preprocessed tweets are fed to XLM-RoBERTa to get the [CLS] embedding and mean-pooled token representations, which represents both global and contextual sentence data. These embeddings are added to the truncated TF-IDF representation and emojis-based features to create a feature representation. The merged vector is then passed through a completely connected classification network comprising of linear layers, ReLU activation, and dropout regularization. There is then a final softmax layer that estimates one of the seven possible categories

Model	Macro-F1
XLM-RoBERTa Only	0.331
XLM-RoBERTa + TF-IDF	0.346
XLM-RoBERTa + TF-IDF + Emoji	0.356

Table 1: Performance of model configurations.

of sentiment: Positive, Negative, Sarcastic, Opinionated, Substantiated, Neutral and None of the Above. We use AdamW optimizer with the learning rate of 2×10^{-5} to train. In order to deal with the imbalance in classes, weighted CrossEntropyLoss computed on inverse class frequencies is employed. Generalization is enhanced by the use of label smoothing. The methods of progressive fine-tuning are followed, i.e. encoder layers are unfreezed across epochs and gradient clipping is utilized to maintain stable optimization.

6 Results and Analysis

6.1 Macro Average F1-Score

The F1 macro average that we employ as needed in the shared task, is the harmonic mean of precision and recall of each class, and then averages all classes in an imbalanced dataset whereby each class is given equal importance.

$$F1_{macro} = \frac{1}{C} \sum \left(\frac{2 \times P_i \times R_i}{P_i + R_i} \right) \quad (2)$$

In which C stands out as the amount of classes and where P_i and R_i represent precision and recall of class i respectively.

6.2 Results

We experimented on three model configurations of our XLM-RoBERTa model. The performance comparison across different model configurations is presented in Table 1. Hybrid feature fusion was effective as the XLM-RoBERTa + TF-IDF + Emoji model did it best. The final model class-wise precision, recall and F1-score is listed in Table 2.

Ablation Analysis:

The addition of TF-IDF features boosts Macro-F1 from 0.331 to 0.346, and emoji features further improve Macro-F1 to 0.356.

The model works well on the class of none and less well on the alternative classes of negative and substantiated. In the shared task, our last system scored 0.3559 in Macro-F1, which ranked 10.

Class	Precision	Recall	F1-score
Negative	0.18	0.22	0.20
Neutral	0.26	0.31	0.28
None	0.91	0.88	0.89
Opinionated	0.45	0.39	0.42
Positive	0.29	0.33	0.31
Sarcastic	0.47	0.44	0.45
Substantiated	0.23	0.26	0.24

Table 2: Class-wise performance of the final model.

6.3 Error Analysis

The model performs well on "None" (F1 = 0.89) but struggles with "Negative" (0.20) and "Substantiated" (0.24). The latter involves factual reasoning that the model is not able to perform, whereas negative sentiment tends to cover the area of sarcasm. Also, sarcasm devoid of emojis and code-mixed text all add to errors. External knowledge and better modeling of sarcasm should be included in future work.

7 Conclusion

We introduce a hybrid approach for Tamil political sentiment analysis, achieving Rank 10 in the shared task. The strategy is a combination of XLM-RoBERTa, TF-IDF, and emoji-based characteristics to manage informal political speech. The model is effective in capturing both semantic and pragmatic cues to multiclass sentiment classification and this reflects the advantage of hybrid feature fusion. The code files of the project can be found on GitHub.[1]

8 Limitations

The dataset is small and imbalanced, limiting the generalizability and influencing the level of performance on minority classes like *Negative* and *Substantiated*. The ambiguities of the political rhetoric and the implicit group mentions also make it more difficult to detect sentiment in political rhetoric (van Dijk, 2002; Coulthard and Johnson, 2007). The model also struggles with implicit sarcasm, where emojis are not available. Although XLM-RoBERTa can offer very strong multilingual representations, other models like MuRIL that are specific to Tamil may be more effective in depicting linguistic nuances.

[1] https://github.com/Inigashree/PoliticalSentiment_DravidianLangTech_2026

References

A. Angdresey, L. Sitanayah, and I. L. H. Tangka. 2025. Sentiment analysis for political debates on youtube comments using bert labeling, random oversampling, and multinomial naïve bayes. *Journal of Computing Theories and Applications*, 2(3):342–354.

Bharathi Raja Chakravarthi, R. Priyadharshini, S. Thavareesan, D. Chinnappa, D. Thenmozhi, E. Sherly, and C. Vasantharajan. 2021. Findings of the sentiment analysis of dravidian languages in code-mixed text. *arXiv preprint arXiv:2111.09811*.

Malcolm Coulthard and Alison Johnson. 2007. Forensic linguistic analysis of political communication. *Journal of Language and Politics*, 6(3):426–431.

S. Sachin Kumar, M. Anand Kumar, and K. P. Soman. 2017. Sentiment analysis of tweets in malayalam using long short-term memory units and convolutional neural nets. In *Proceedings of the 5th International Conference on Mining Intelligence and Knowledge Exploration (MIKE 2017)*, pages 320–334. Springer.

B. Prathvi, K. Manavi, K. Subrahmanyapoojary, Asha Hegde, G. Kavya, and Hosahalli Shashirekha. 2024. MUCS@dravidianlangtech-2024: A grid search approach to explore sentiment analysis in code-mixed tamil and tulu. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 257–261, St. Julian's, Malta. Association for Computational Linguistics.

R. Rajalakshmi, S. Selvaraj, A. Shibani, and B. R. 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.

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.

Stefan Stieglitz and Linh Dang-Xuan. 2012. Political communication and influence through microblogging—an empirical analysis of sentiment in twitter messages. *Journal of e-Democracy*.

Andranik Tumasjan, Timm O. Sprenger, Philipp G. Sandner, and Isabell M. Welpe. 2010. Predicting elections with twitter: What 140 characters reveal about political sentiment. *Journal of Information Technology*.

Teun A. van Dijk. 2002. Political discourse and ideology. *Discourse & Society*, 13(2):203–221.

Mani Vegupatti, Kishore Kumar Ponnusamy, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Durairaj Thenmozhi, Prasanna Kumar Kumaresan, and Sathiyaraj Thangasamy. 2026. TamilPoliSent 2026: A Shared Task report on Multiclass Political Sentiment Analysis in Tamil. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

TAMILGOODBADTXT@DravidianLangTech 2026:A Multilingual Transformer-Based Approach for Abusive Language Identification in Tamil Social Media

K Varalakshmi[1], **B Bharathi**[2]
[1] St. Joseph's Institute of Technology, Tamil Nadu, India
[2]Sri Sivasubramaniya Nadar College of Engineering, Tamil Nadu, India
varaluckky.2@gmail.com, bharathib@ssn.edu.in

Abstract

It is difficult to detect abusive language, particularly in social networks for low-resource languages like Tamil. Spelling errors, informal expressions and code-mixing make it even more challenging to read text from social media. The current work proposes a multilingual transformer-based approach to detect abusive content in Tamil text. A pretrained XLM-RoBERTa model is used to learn contextual and semantic representations from the input text. This is a general pipeline comprising preprocessing, tokenization, and binary classification (abusive / non-abusive). Experiments are performed on Tamil social media datasets that have abusive and non-abusive data. The results reveal that multilingual transformer models achieve good performance in low-resource scenarios. The proposed model attains an F1-score of 78.64%, which shows the potential of using cross-lingual pretrained models for the detection of abusive Tamil language.

Keywords: Abusive language detection; Tamil NLP; multilingual transformers; XLM-RoBERTa; Low-resource languages.

1 Introduction

Social network sites have seen unprecedented user-generated content, which has seen individuals express their views, feelings, and thoughts on the Internet under the freedom of expression. This has contributed to the rapid spread of abusive, violent, and hate speech as much as it has promoted digital communication, and this is a very big threat to man and society at large. Therefore, the automatic detection of the existence of abusive language has become a significant concern in the natural language processing field that has significance in offering content regulation and in offering a safer Internet experience. Tamil is a low-resource Dravidian language, and the fact that the abusive text is identified in Tamil social media presents certain challenges. Online Tamil is very informal and normally contains a variation of spelling and dialect and is highly code-mixed with English. These characteristics render the conventional rule-based systems as well as the conventional machine learning techniques to be less efficient, in part due to the fact that there are no huge annotated data sets. As such, it would not be an easy task to create effective Tamil abusive language detectors. The more recent progress in deep learning, particularly transformer-based language models, has demonstrated itself to be able to perform state-of-the-art in a large variety of text classification tasks. Rich multilingual cross-lingual contextual representations Multilingual transformer models are trainable on large-scale multilingual corpora with rich cross-lingual contextual representations. In this case, low-resource languages, such as the XLM-RoBERTa model specifically, are especially suitable to transfer linguistic information of the high-resource languages and improve learning of representation and classification performance. This study addresses the challenge of detecting abusive language in the text of Tamil social media through multilingual transformer-based models. The proposed system trains a multilingual model that is already trained on Tamil-labeled data to identify the texts as abusive or not. Such semantic relationships and contextual clues that are involved with the linguistic variation and informal style of writing that is evident in Tamil social media can be dealt with successfully by the system. The results of the experiment demonstrate that the use of multilingual transformer models is efficient in the condition of the absence of the data and language barriers, and they may be considered a viable option in the situation with the abusive language detection in the low-resource languages(i.e Tamil)

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 392–396
July 3, 2026 ©2026 Association for Computational Linguistics

2 Literature Survey

The detection of abusive language has also received a lot of focus particularly in low-resource and multilingual environment. The initial methods used were based on the traditional machine learning and deep neural networks like CNNs and RNNs to detect hate speech in social media textual data (Zhang et al., 2018). Later on, their performance was enhanced by transformer-based models such as BERT which is able to capture contextual semantics and thus enhance its performance over others (Mozafari et al., 2019). OffensEval shared task focused on the detection of types of offensive contents and targets (Zampieri et al., 2019).

As multilingual transformers emerged, studies were expanded to low resource languages. Rajalakshmi et al. showed the efficiency of Tamil multilingual models to detect abusive comments (Rajalakshmi et al., 2022). Recent works at DravidianLangTech workshops included Tamil and Malayalam on pretrained transformers, such as detecting women-targeted abusive text in a series of works on the topic, including (Thirumoorthy et al., 2025; Al Nahian et al., 2025; Bade et al., 2025).

Multilingual toxicity detection: Multilingual hybrid models based on XLM-RoBERTa with CNN and BiLSTM have demonstrated encouraging outcomes (Singhal et al., 2025). Moreover, cross-lingual generalization was also enhanced by large language model-based approaches (Usman et al., 2025). The article by Nalini et al. offers an overview of the latest innovations (Nalini et al., 2024). The previous research article by Gamback and Sikdar, as well as Davidson et al. (Gambäck and Sikdar, 2017; Davidson et al., 2017), indicated that CNNs have been proven to be effective and that differentiating between hate speech and offensive language proves to be challenging.

Transformer-based models also enhanced the detection of abusive language. Das et al. (Das et al., 2021) and Mishra et al. (**?**) demonstrated the superiority of BERT, mBERT, and XLM-R when it comes to a multilingual and low-resource setting. Studies of Dravidian languages in recent times still utilize pretrained transformers of Tamil and Malayalam (Thirumoorthy et al., 2025; Al Nahian et al., 2025).

The results of the article by (Rajiakodi et al., 2026) reveal that the issues that were encountered include morphological richness and the necessity of powerful classification models in the Tamil language. In spite of this, such problems as linguistic diversity, code-mixing, and annotated data are still present. Out of these challenges, this paper introduces a transformer-based multilingual model that is fine-tuned on the detection of Tamil abusive language.

3 Proposed Work

In this work, the input data consists of Tamil text and a fine-tuned multilingual transformer model is used to capture the contextual and semantic representations to detect abusive language. The system is transformer-based Where raw Tamil text is provided as input and the outputs are binary classification. Model agnostic, enabling a variety of pretrained transformer backbones to be used. It includes pre-processing of text, generation of contextual embedding and classification. The complete architecture of the proposed system is illustrated in Figure 1.

3.1 Input Dataset Representation

A corpus of Tamil social media posts in CSV format was fed into the system as input data. There are 3,652 samples in the training set, categorized into two classes: Abusive and Non-Abusive. The test data consists of 913 examples, which are used for testing. Each example sentence is a Tamil sentence without any metadata. To ensure linguistic diversity, no manual processing is involved in the process.

3.2 Text Encoding and Tokenization

Each of the input sentences is segmented into a series of subword tokens using a model-specific tokenizer.Each sentence in the input is split into a sequence of subword tokens, using a model-specific tokenizer. This process creates input token IDs, attention masks and pads to a fixed sequence length. The method proved to be effective in capturing the variations in the morphology and the code-mixing expressions in the Tamil social media text.

3.3 Transformer-Based Feature Extraction

A pretrained transformer encoder is used to encode the text into deep contextual representations. These are the multilingual transformer models that are tested:

- XLM-RoBERTa Base
- IndicBERT
- MuRIL

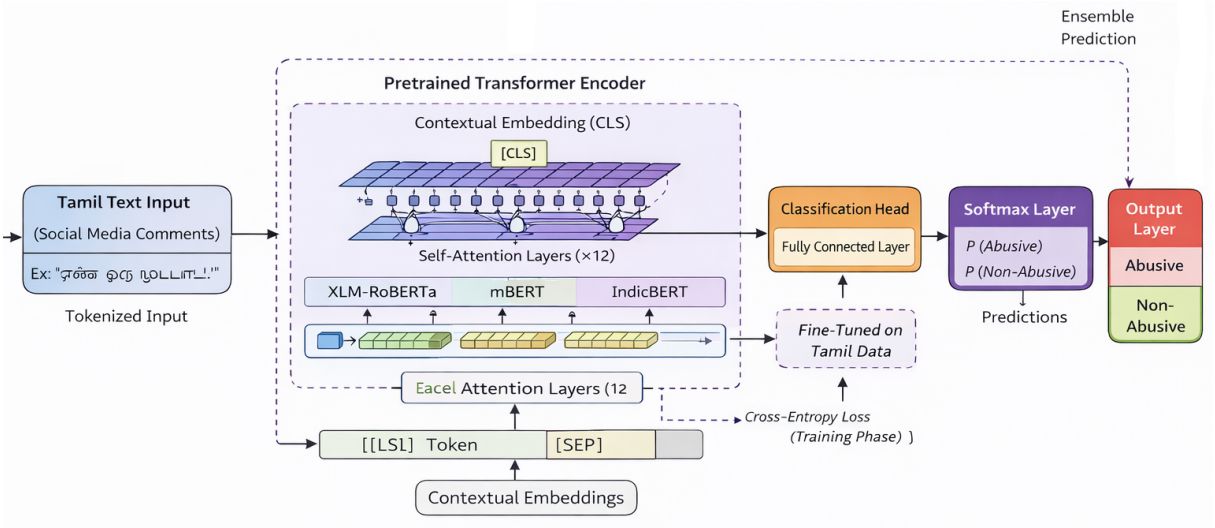


Figure 1: Proposed transformer-based architecture for abusive language detection in Tamil text

Multi-head self-attention is adopted in these models to model long-range dependency and the semantic context. The pretrained representations are fine-tuned to better suit the abusive language detection task.

3.4 Classification Layer

The contextual embeddings produced by the transformer encoder are fed to a classification head comprising a fully connected neural network with a SoftMax activation function. The model is predicting probabilities for the two classes Abusive and Non-Abusive. The system is trained with cross entropy loss function and optimized with the AdamW optimizer.

3.5 Output Generation

In inference, the trained model is used to predict the class label for the unseen test data. The expected results are then turned into the final class labels that are evaluated with standard performance measures.

4 Dataset

The data set this work uses is for detection of abusive language in the Tamil social media comments. It is split into two sets: a training set and a test set in order to perform supervised learning and evaluation. It has 3652 sentences in the training set and 913 sentences in the test set, totaling 4565 sentences.

The dataset is provided as part of a shared task on Tamil abusive language detection. All training samples are manually annotated as **Abusive** or **Non-Abusive**. The dataset reflects real-world social media characteristics, including informal language, spelling variations, colloquial expressions, emojis, and code-mixing with English.

Table 1: Dataset Statistics

Dataset Split	Samples	Annotation
Training	3,652	Abusive / Non-Abusive
Test	913	Unlabeled
Total	**4,565**	–

The training data is imbalanced, with **Non-Abusive** as the majority class and **Abusive** as the minority class, reflecting real-world distribution patterns.

The source code GitHub link is available here: [1].

5 Experimental Set Up And Results Analysis

The model training was carried out on a GPU environment with the aim of efficient convergence and computational performance improvement. A standard preprocessing pipeline was applied to the input text to ensure uniformity in the data, which includes normalization, tokenization with transformer-based tokenizers, padding, truncation to the maximum sequence length, and creation of attention masks needed by the model. The objective function for training the classification problem (abusive vs not abusive) was binary cross-entropy

[1] https://github.com/Varalakshmi2793/TAMILGOODBADTXT.git

loss. Optimization was carried out with Adam optimiser with the proper learning rate schedule to guarantee a stable convergence. The model was trained end-to-end to acquire multilingual contextual representations and semantics and the syntactic properties of Tamil social media texts.

Model trained was tested against 913 unseen samples to assess the proposed system. Standard classification metrics such as Accuracy, Precision, Recall, Macro F1-Score, and Weighted F1-Score were used to measure the performance. Accuracy is the overall correctness of predictions. Precision is the ratio of correctly predicted samples that are abusive to all samples predicted as abusive. Recall is the model's ability to accurately detect real abusive content. The F1-Score gives both the Precision and the Recall a balance. Macro F1-Score treats both classes equally and Weighted F1-Score considers the class imbalance.

The quantitative performance of the proposed model (RUN 1) is presented in Table 2.

Table 2: Performance Evaluation Metrics

Metric	Value
Accuracy	0.7864
Precision	0.7881
Recall	0.7876
Macro F1-Score	0.7864
Weighted F1-Score	0.7800

It is found that, the model has an overall accuracy of 78.64%. The accuracy of 78.81% implies that the model is able to identify abusive content with relatively low amount of false positive and the recall value of 78.76% shows that maximum number of abusive instances are identified. The Macro F1-Score of 78.64% shows that the model performs well on both classes, and the Weighted F1-Score of 78.00% takes into account the impact of class imbalance. The model does not show strong class bias, with the close values of precision and recall. Additional information about the classification behavior is presented in the confusion matrix shown in Figure 2.

Based on the classification results in the 913 test samples, the model achieved True Negatives (TN) = 359, False Positives (FP) = 98, False Negatives (FN) = 97 and True Positives (TP) = 360. The results show that a good percentage of the samples were classified correctly, far more than misclassified samples.

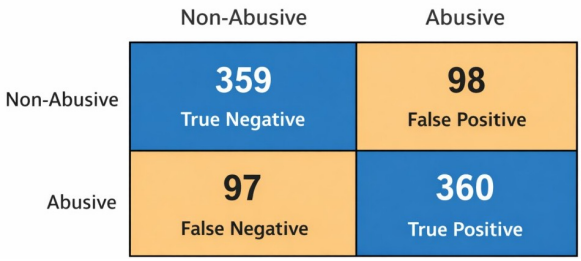


Figure 2: Confusion matrix of the proposed abusive language detection model

The near equivalence of the false positive and false negative rates suggest that the model is not heavily skewed towards either of the two classes of abusive or non-abusive. Ambiguous language, sarcasm, implicit abusive language, informal language and code-mixed Tamil-English text often used on social media platforms are some of the reasons for misclassification. In general, the multilingual transformer based model is effective for detecting abusive language in low resource setting of Tamil language and has good generalization capability.

6 Conclusion

The work proposed a multilingual transformer based approach for Tamil abusive language detection. The model was trained on annotated Tamil text and evaluated on 913 test samples. The proposed system achieved 78.64% accuracy with balanced value of precision, recall and F1 score. The results show that transformer-based contextual embeddings are capable of capturing the semantic and syntactic properties of Tamil text, and hence they can effectively detect abusive language in low-resource settings. The analysis of the confusion matrix also shows a stable classification performance without a significant bias to any class. Future work could focus on incorporating bigger domain specific datasets, handling code-mixed and sarcastic texts more efficiently and exploring ensemble transformer architectures to improve the performance further.

References

Abdullah Al Nahian, Mst Rafia Islam, Azmine Toushik Wasi, and Md Manjurul Ahsan. 2025. Nlpopsciol@ dravidianlangtech 2025: Classification of abusive tamil and malayalam text targeting women using pretrained models. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 38–45.

Girma Yohannis Bade, Zahra Ahani, Olga Kolesnikova, José Luis Oropeza, and Grigori Sidorov. 2025. Gs_dravidianlangtech@ 2025: Women targeted abusive texts detection on social media. *arXiv preprint arXiv:2504.02863*.

Mithun Das, Somnath Banerjee, and Punyajoy Saha. 2021. Abusive and threatening language detection in urdu using boosting based and bert based models: A comparative approach. *arXiv preprint arXiv:2111.14830*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using convolutional neural networks to classify hate-speech. In *Proceedings of the first workshop on abusive language online*, pages 85–90.

Marzieh Mozafari, Reza Farahbakhsh, and Noel Crespi. 2019. A bert-based transfer learning approach for hate speech detection in online social media. In *International conference on complex networks and their applications*, pages 928–940. Springer.

C Nalini, R Shanthakumari, Y Agashia Maria, T Janarthanan, and M Manibharathi. 2024. Advancements in offensive language detection: A comprehensive review and experimental analysis. *Journal of Information Assurance & Security*, 19(4).

Ratnavel Rajalakshmi, Ankita Duraphe, and Antonette Shibani. 2022. Dlrg@ dravidianlangtech-acl2022: Abusive comment detection in tamil using multilingual transformer models. In *Proceedings of the second workshop on speech and language technologies for Dravidian languages*, pages 207–213.

Saranya Rajiakodi, Shunmuga Priya Muthusamy Chinnan, Rajalakshmi R., Kathiravan Pannerselvam, Bhuvaneswari Sivagnanam, Jananayagan V, Charmathi Rajkumar, R Ramesh Kannan, and Bharathi Raja Chakravarthi. 2026. From Comments to Harm: A Findings Report on Abusive Tamil Text Targeting Women on Social Media Shared Task- DravidianLangTech@ACL 2026. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Nikita Singhal, Avadhesh Yadav, Ankush Ankush, Giriraj Singh, and Ronak Kumar. 2025. Leveraging xlm-roberta with cnn and bilstm for hinglish toxicity detection. *Journal of Communications Software and Systems*, 21(4):394–403.

Shanmitha Thirumoorthy, Thenmozhi Durairaj, and Ratnavel Rajalakshmi. 2025. Hydrangea@ dravidianlantech2025: Abusive language identification from tamil and malayalam text using transformer models. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 580–584.

Muhammad Usman, Muhammad Ahmad, Grigori Sidorov, Irina Gelbukh, and Rolando Quintero Tellez. 2025. A large language model-based approach for multilingual hate speech detection on social media. *Computers*, 14(7):279.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. 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 1415–1420.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European semantic web conference*, pages 745–760. Springer.

TamilVoiceLab@DravidianLangTech 2026: Investigating Whisper Tamil Large-v2 for Dialectal Tamil Speech Recognition

S.B.PRIYA
St. Joseph's Institute of Technology
Chennai, India
priyait0843@gmail.com

B.BHARATHI
Sri Sivasubramaniya Nadar College of Engineering
Chennai, India
bharathib@ssn.edu.in

Abstract

Automatic Speech Recognition (ASR) for languages rich in dialects and those with limited resources presents significant challenges due to the variations in pronunciation and vocabulary across different regions. This study offers a baseline evaluation of the Whisper Tamil Large-v2 model without fine-tuning for the shared task of Tamil Dialect Speech Recognition. The focus is on the ASR subtask, utilising dialectal Tamil speech recordings gathered from various regional dialects within Tamil Nadu. The pretrained Whisper Tamil Large-v2 model was assessed directly, without any supplementary fine-tuning or domain adaptation. A total of 579 dialect speech samples were used for experimentation, with performance evaluated based on Word Error Rate (WER). The model recorded a WER of 0.71, indicating that even robust multilingual pretrained models encounter challenges in dialect-rich and low-resource environments. These findings underscore the necessity for dialect-aware adaptation and the importance of balanced dialect training data to develop effective Tamil ASR systems.

Keywords: Tamil ASR, Dialect Speech Recognition, Whisper, Low-Resource Speech Processing, Zero Fine-Tuning

1 Introduction

ASR, or automatic speech recognition, uses computer models to translate speech into text. ASR performance has significantly improved for high-resource languages like Mandarin and English thanks to deep learning and transformer-based models. For low-resource languages and dialect-rich environments with little training data, ASR is still difficult.More than 75 million individuals speak Tamil, a significant Dravidian language. Northern, Southern, Western, and Central Tamil Nadu all have strong regional dialects. Pronunciation, rhythm, vocabulary, and sentence structure vary among these dialects. Speech from one area may sound significantly different from speech from another due to this diversity. Such dialect discrepancies are a common problem for ASR systems that were mostly trained on standard Tamil. Strong cross-lingual abilities have recently been demonstrated by multilingual models such as Whisper. Whisper can do recognition without task-specific fine-tuning because it was trained on vast amounts of multilingual speech samples. Its performance on unadapted dialectal Tamil speech, however, has not been thoroughly investigated. Whisper Tamil Large v2 is assessed in this study using dialectal Tamil ASR with zero fine-tuning. 580 dialect voice samples that were captured in natural settings are used to directly evaluate the model. Word Error Rate, which contrasts predicted text with ground truth transcription, is used to gauge performance. With a Word Error Rate of 0.71, the model demonstrated a notable decline in performance under dialect-rich circumstances. The model struggles with intra-linguistic variances present in Tamil dialect speech, despite having extensive multilingual training. These results emphasise the necessity of fine-tuning low-resource models with balanced training data, dialect-aware fine-tuning, and parameter-efficient adaptation strategies. In addition to supporting future research toward more inclusive and regionally resilient speech systems, this work establishes a baseline for enhancing Tamil dialect ASR.

2 Related Work

Recent research indicates strong growth in Tamil speech recognition, particularly in dialectal and low-resource settings. Many studies focus on improving data quality, adapting large models like Whisper, and handling real-world challenges such as code switching and speaker variation.

Low Rank Adaptation has recently been applied

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 397–402
July 3, 2026 ©2026 Association for Computational Linguistics

to Whisper for inclusive Tamil speech recognition. This method fine-tuned Whisper efficiently while reducing computational cost and improving recognition for diverse and vulnerable speaker groups (Acharya et al., 2025). Code switching remains a major challenge in end-to-end ASR, especially for Indian languages such as Tamil, where mixed language speech is common. A systematic review explained the difficulties in handling code-mixed speech and highlighted model adaptation strategies for better performance (Agro et al., 2025). Large multilingual speech models have been shown to improve low-resource recognition through cross-lingual transfer. XLS R demonstrated that training on many languages enhances recognition accuracy for dialect-rich languages (Babu et al., 2021). Self-supervised learning has played a major role in advancing ASR. The wav2vec 2.0 framework showed that learning speech representations from raw audio before fine-tuning significantly improves performance for low-resource languages (Baevski et al., 2020). Recent work has also explored multilingual speech-to-speech translation using unified end-to-end architectures. These systems combine recognition and translation within a single framework (Beltrán Lobato, 2025). Dialect-rich corpora are important for robust Tamil ASR. A multi-dialect Tamil speech corpus demonstrated that including diverse regional accents in training data improves recognition robustness (Bharathi et al., 2025). Integrating external language models with Whisper has been shown to enhance low-resource recognition. Whisper LM reduced recognition errors by strengthening language modelling alongside acoustic modelling (de Zuazo et al., 2025). Fine-tuning Whisper for low-resource languages has produced strong improvements. Adaptation experiments on Amharic confirmed that even limited domain data can significantly enhance performance (Gete et al., 2025). Benchmarking is essential for the fair evaluation of Indian language speech systems. IndicSuperb introduced a standardised evaluation framework for speech processing tasks across Indian languages (Javed et al., 2023). Different fine-tuning strategies for Whisper have been compared in low-resource ASR settings. Targeted adaptation methods,, such as parameter-efficient tuning,, significantly improved recognition accuracy (Liu et al., 2024). Multilingual adaptation techniques have also been studied for RNN-based ASR systems. Knowledge transfer among languages was shown to improve recognition accuracy (Miiller et al., 2018). Whisper has been adapted for Indian language speech tasks beyond recognition, including sentiment analysis and speaker diarization. These studies confirm that proper fine-tuning enables effective adaptation to Indian languages (Papala et al., 2023). Transformer-based ASR models have been successfully developed for other Dravidian languages such as Kannada. These findings support the effectiveness of transformer encoders for speech recognition in related languages like Tamil (Prasad et al., 2026). Whisper, trained with large-scale weak supervision, demonstrated strong robustness to noise, accents, and spontaneous speech across multiple languages (Radford et al., 2023). Cross-lingual generalisation and neuron sharing have been studied in multilingual models. Combining shared and language-specific representations improves performance in low-resource settings (Riemenschneider and Frank, 2025). Real-time Tamil dialect speech recognition has also been explored. A system combining recognition and summarisation showed the importance of dialect-aware modelling for practical applications (Saranya et al., 2025). Multilingual Indian speech translation corpora support cross-task evaluation of speech systems. Such resources highlight the growing need for multilingual benchmarking in Indian languages (Shah et al., 2025). Neural models have been used for dialect recognition using convolutional networks and language embeddings. These models effectively distinguish closely related dialects (Shon et al., 2018). Sequence-to-sequence ASR systems have been combined with Indic large language models for speech translation. Integrating stronger language models improves overall system performance (Wei et al., 2025). Large language models and multilingual training strategies have also been explored for low-resource speech recognition. Transfer learning was shown to be effective in improving performance under limited data conditions (Zhang and Huang, 2025). Overall, recent literature shows three major trends. First, dialect-specific and diverse datasets improve ASR quality. Second, fine-tuning large pretrained models like Whisper reduces error rates in low-resource conditions. Third, multilingual and cross-lingual learning helps transfer knowledge across related languages. These studies strongly support the need for adapting Whisper-based models for Tamil dialect recognition, especially when training data is limited.

Table 1: Tamil Dialect Speech Dataset Overview

Category	Details
Language	Tamil
Number of Dialects	4
Dialects Included	Northern, Southern, Western, Central
Dialect Regions	Regions across Tamil Nadu
Speech Type	Spontaneous and Read Speech
Sampling Rate	16 kHz
Recording Environment	Natural acoustic conditions
Training Duration	9.22 hours
Test Duration	2.05 hours
Test Samples Used	580 audio files
Evaluation Metric	Word Error Rate

3 Dataset Description

A shared job on Tamil voice processing provided the data for this investigation. The Tamil Dialect Speech dataset used in this work was released through the DravidianLangTech shared task (Bharathi et al., 2026). The dataset includes speech recordings from four dialect regions of Tamil Nadu and supports both dialect identification and speech recognition research. Speech recordings from four dialects—Northern, Southern, Western, and Central—are included in the corpus. There is a corresponding Tamil transcription for every training sample. 5,134 recordings totalling roughly 9.22 hours make up the training set. Although the distribution among dialects is not entirely balanced, the data shows regional variance and natural pronunciation. There are 579 unlabeled audio samples in a different test set. The review is objective because the transcripts are not accessible. Table 1 provides a fair comparison across systems and enumerates the dataset's key features.

4 Proposed Methodology

The proposed system uses Whisper Tamil Large v2 for dialectal Tamil speech recognition. The main objective is to convert Tamil speech audio into text using a pretrained multilingual Automatic Speech Recognition model. No additional training or fine tuning was performed in this work. Instead, the study evaluates how well a large pretrained model performs directly on dialect rich Tamil speech.

The speech recordings were obtained from the shared task dataset and maintained at a sampling rate of 16 kHz, which matches the requirement of the Whisper model. Each audio waveform was processed using the Whisper processor to extract log Mel spectrogram features. These features capture important acoustic information such as frequency patterns and energy variations present in the speech signal.

The extracted features were then passed into the Whisper encoder. The encoder learned hidden speech representations related to pronunciation, rhythm, accent, and speaking style. These learned representations were used by the decoder to generate Tamil text in an end to end manner. Beam search decoding was applied to improve prediction accuracy by selecting the most probable sequence of words.

All test samples were processed using the same inference pipeline. The generated transcriptions were stored for analysis and reproducibility.

System performance was evaluated using Word Error Rate. The model achieved a WER of 0.71, showing that dialectal Tamil speech remains challenging in a zero fine tuning setting. Most recognition errors occurred in region specific vocabulary, colloquial expressions, and pronunciation variations across dialects.

Even though the error rate was high, the model was able to generate partially understandable Tamil text and correctly identify many common words. This indicates that large pretrained multilingual speech models possess reasonable generalization capability even without dialect specific adaptation. However, the results also highlight the importance of dialect-aware fine-tuning for robust Tamil speech recognition. Figure 1 illustrates the

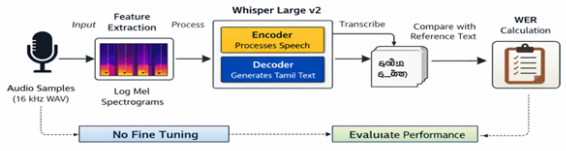


Figure 1: Proposed Tamil Dialect ASR System Architecture

overall architecture of the proposed Tamil dialect speech recognition system. The system accepts Tamil speech audio in 16 kHz WAV format as input. The audio signal is first converted into log Mel spectrogram features using the Whisper processor. These features capture important acoustic patterns from the speech signal and are then passed to the pretrained Whisper Tamil Large v2 encoder-decoder model for Tamil text generation.

The generated transcriptions are finally evaluated using Word Error Rate.

5 Evaluation and Results

Standard metrics for voice recognition are used to assess this system. Measuring the degree to which the predicted text differs from the original reference text is the primary objective. **Word Error Rate** Word Error Rate, or WER, is the most common metric in ASR. It measures errors at the word level. WER is calculated as:

$$WER = \frac{S + D + I}{N} \quad (1)$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions, and N is the total number of words in the reference transcription. Substitution means one word is replaced by another word. Deletion means a word is missing. Insertion means an extra word is added. A lower WER means better performance. In this system, the obtained WER is: [WER = 0.71] This means about 71 per cent of the words are incorrect when compared to the reference transcript. **Sample Model Outputs** Below are sample transcriptions generated by the system:

File number	Predicted output	Target Output
test_0006	செய்து என் பையனுக்கு பிடிச்சது பிரியாணி தான் சிக்கன் பிரியாணி அதனால வரத்து ஒருநாள் எப்படியாவது செஞ்சு கொடுப்போம்	அதிகமா செய்றது, அவனுக்கு ஏன் பையனுக்கு பிடிச்சது பிரியாணி தான் சிக்கன் பிரியாணி தான். அதனால வாரத்துல ஒரு நாள் எப்படியாவது அவனுக்கு செஞ்சு கொடுப்போம்.
test_0015	நான் வந்து ஜீ தமிழ் தான் ரொம்ப பாப்பேன் எனக்கு தான் ரொம்ப பிடிக்கும்	நான் வந்து ஜீ தமிழ் தான் ரொம்ப பாப்பேன். எனக்கு அதுதான் ரொம்ப புடிக்கும்.
test_0113	பொழுது அனைக்கும் கேம் தான் விளையாடு நீ என்ன வந்து பேசுற	பொழுதன்னைக்கும் கேம் தான வெளையாடு நீ என்ன வந்து பேசுற

Figure 2: Comparison of Predicted and Target Transcriptions

Figure 2 shows the sample output generated by the proposed Tamil ASR system along with the target transcription. The highlighted words indicate recognition errors. Most errors occur in colloquial expressions and regional speech patterns. Table 2 compares the proposed system with recent Tamil speech recognition systems reported in the literature. The comparison is based on Word Error Rate, where lower values indicate better recognition performance. The proposed system achieved a WER of 0.71 using Whisper Tamil Large v2 without any fine tuning. Compared with other systems that use larger datasets and adaptation techniques, the results highlight the challenges of dialectal Tamil ASR in low resource settings. This comparison provides a baseline for future improvement in Tamil speech recognition systems.

Table 2: Comparison of Tamil ASR Systems

Paper / System	Model Used	Fine Tuning	Dataset Type	Reported WER	Notes
Real-time Continuous Tamil Dialect Speech Recognition and Summarisationn (2025)	Custom Dialect ASR Model	Yes	Multi-dialect Tamil speech	0.35–0.45	Trained specifically on dialect speech
Multi-Dialect Speech Corpus Creation for Enhancing Tamil ASR (2025)	Dialect-specific ASR	Yes	Tamil dialect corpus	0.40	Focused on corpus development
Junlp@LT-EDI 2025 Efficient Low-Rank Adaptation of Whisper	Whisper with LoRA	Yes	Inclusive Tamil speech	0.38	Used parameter-efficient fine tuning
Exploration of Whisper Fine-Tuning Strategies for Low-Resource ASR (2024)	Whisper	Yes	Low-resource languages, including Tamil	0.30–0.45	Fine-tuning improves results clearly
This System: Whisper Tamil Large-v2	Whisper Tamil Large-v2	No	Tamil dialect speech (580 samples)	0.71	Zero fine-tuningng baseline

Result The system was tested on 579 dialect voice samples using Whisper Tamil Large v2 with zero fine-tuning. The Word Error Rate was high at 0.71 because no domain adaptation was used. Nevertheless, the model generated comprehensible Tamil text, demonstrating that huge pretrained models may function fairly well in environments with limited resources. Better training data and dialect-specific tuning can lead to further advances. Code for the proposed system is available in the link ([1])

6 Conclusion

This study evaluated Whisper Tamil Large v2 for Tamil dialect speech recognition using 579 speech samples in a zero fine-tuning setting. The model achieved a Word Error Rate of 0.71, showing that dialect-rich Tamil speech remains challenging for pretrained multilingual ASR systems. Most recognition errors occurred in region-specific vocabulary, colloquial expressions, and pronunciation variations. Even so, the model generated partially understandable Tamil text and correctly identified many common words, showing reasonable generalisation capability without dialect-specific adaptation. This work provides a baseline for dialectal Tamil ASR and highlights the importance of dialect-aware fine-tuning and balanced training data for improving speech recognition performance in low-resourced dialect settings.

[1] https://github.com/Prisur2013/TAMILVOICELABS_1991/tree/main

References

Priyobroto Acharya, Soham Chaudhuri, Sayan Das, Dipanjan Saha, and Dipankar Das. 2025. Junlp@ lt-edi-2025: Efficient low-rank adaptation of whisper for inclusive tamil speech recognition targeting vulnerable populations. In *Proceedings of the 5th Conference on Language, Data and Knowledge: Fifth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 17–25.

Maha Tufail Agro, Atharva Kulkarni, Karima Kadaoui, Zeerak Talat, and Hanan Aldarmaki. 2025. Code-switching in end-to-end automatic speech recognition: A systematic literature review. *arXiv preprint arXiv:2507.07741*.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick Von Platen, Yatharth Saraf, Juan Pino, and 1 others. 2021. Xls-r: Self-supervised cross-lingual speech representation learning at scale. *arXiv preprint arXiv:2111.09296*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Érik Beltrán Lobato. 2025. Multilingual speech-to-speech machine translation.

B. Bharathi, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, S. Saranya, and S. Suhasini. 2026. Findings in Tamil Dialect Speech Recognition and Classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

B Bharathi, S Saranya, P Vijayalakshmi, and T Nagarajan. 2025. Multi-dialect speech corpus creation for enhancing tamil automatic speech recognition. *Circuits, Systems, and Signal Processing*, pages 1–19.

Xabier de Zuazo, Eva Navas, Ibon Saratxaga, and Inma Hernáez Rioja. 2025. Whisper-lm: Improving asr models with language models for low-resource languages. *arXiv preprint arXiv:2503.23542*.

Dawit Ketema Gete, Bedru Yimam Ahmed, Tadesse Destaw Belay, Yohannes Ayana Ejigu, Sukairaj Hafiz Imam, Alemu Belay Tessema, Mohammed Oumer Adem, Tadesse Amare Belay, Robert Geislinger, Umma Aliyu Musa, and 1 others. 2025. Whispering in amharic: Fine-tuning whisper for low-resource language. *arXiv preprint arXiv:2503.18485*.

Tahir Javed, Kaushal Bhogale, Abhigyan Raman, Pratyush Kumar, Anoop Kunchukuttan, and Mitesh M Khapra. 2023. Indicsuperb: A speech processing universal performance benchmark for indian languages. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 12942–12950.

Yunpeng Liu, Xukui Yang, and Dan Qu. 2024. Exploration of whisper fine-tuning strategies for low-resource asr. *EURASIP Journal on Audio, Speech, and Music Processing*, 2024(1):29.

Markus Miiller, Sebastian Stiiker, and Alex Waibel. 2018. Multilingual adaptation of rnn based asr systems. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5219–5223. IEEE.

Gowtham Papala, Aniket Ransing, and Pooja Jain. 2023. Sentiment analysis and speaker diarization in hindi and marathi using using finetuned whisper: sentiment analysis in hindi and marathi. *Scalable Computing: Practice and Experience*, 24(4):835–846.

Chandrika Prasad, Veenga Gode Swamy Rao, R China Appala Naidu, and 1 others. 2026. An asr transformer-based model for kannada speech-to-text transcription. *Journal of Artificial Intelligence and Technology*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Frederick Riemenschneider and Anette Frank. 2025. Cross-lingual generalization and compression: from language-specific to shared neurons. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13470–13491.

S Saranya, B Bharathi, S Gomathy Dhanya, and Aishwarya Krishnakumar. 2025. Real-time continuous tamil dialect speech recognition and summarization. *Circuits, Systems, and Signal Processing*, 44(4):2855–2881.

Sanket Shah, Kavya Ranjan Saxena, Kancharana Manideep Bharadwaj, Sharath Adavanne, and Nagaraj Adiga. 2025. Indicst: Indian multilingual translation corpus for evaluating speech large language models. In *2025 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pages 1–5. IEEE.

Suwon Shon, Ahmed Ali, and James Glass. 2018. Convolutional neural networks and language embeddings for end-to-end dialect recognition. *arXiv preprint arXiv:1803.04567*.

Xuchen Wei, Yangxin Wu, Yaoyin Zhang, Henglyu Liu, Kehai Chen, Xuefeng Bai, and Min Zhang. 2025. Hitsz's end-to-end speech translation systems combining sequence-to-sequence auto speech recognition model and indic large language model for iwslt 2025 in indic track. In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, pages 405–411.

Jiqiao Zhang and Degen Huang. 2025. Speech recognition for low-resource languages using large language models and related-language data. In *International Conference on Image, Signal Processing, and Pattern Recognition (ISPP 2025)*, volume 13664, pages 1264–1269. SPIE.

Team Oryu@DravidianLangTech 2026: A Multilingual Transformer Approach for Hope Speech Detection in Code-Mixed Tulu

Joyeta Barua Moni, Noore Tamanna Orny, Md. Abtahee Kabir, Hasan Murad
Department of Computer Science and Engineering,
Chittagong University of Engineering and Technology, Bangladesh
{u2104130, u2104092, u2104089}@student.cuet.ac.bd, hasanmurad@cuet.ac.bd

Abstract

Hope speech detection appears to have an essential role to play in fostering positive and inclusive communication on social media, especially in low-resource multilingual settings. This paper describes the system submitted by Team Oryu for Task 1: Coarse-Grained Hope Tone Classification in Code-Mixed Tulu. The task involves classifying comments in social media texts into one of the four classes: Encouraging, Discouraging, Uninvolved, and Blended Tone. The texts in this task show heavy code-mixing between Tulu, English, and Kannada. In order to overcome this challenge, we employed a fine-tuned multilingual transformer model, code-mixed text processing, data augmentation, and class-weighted loss to handle class imbalance. Our proposed system achieved a Macro F1-score of 63%, securing 3rd position on the shared task. The results demonstrate the efficacy of multilingual transformer models in emotionally nuanced classification in code-mixed environments while underscoring the difficulties in capturing blended emotional tones.

1 Introduction

Hope speech detection has emerged as a new direction of research in Natural Language Processing (NLP), especially with respect to creating a positive online discourse and promoting mental well-being. Although a significant amount of research has focused on detecting negative content such as hate speech, abusive language, and offensive expressions, comparatively little work has been done on identifying positive and supportive communication. Hope speech detection aims to identify language that conveys encouragement, optimism, and empathy.

The shared task presents the first benchmark for hope speech detection in code-mixed Tulu. Tulu, a low-resource Dravidian language mainly spoken in the coastal districts of Karnataka and Kerala, has received limited attention in NLP research. On social media platforms such as YouTube, Tulu speakers frequently use code-mixed text consisting of Tulu, English, and Kannada, often written in Roman script.

The first task of the shared task focuses on the classification of coarse-grained hope tones, where each comment is categorized into one of four classes: *Encouraging*, *Discouraging*, *Uninvolved*, and *Blended Tone*. This task differs from conventional sentiment classification as it requires a more nuanced understanding of emotional tone.

To address this challenge, we propose a transformer-based approach for coarse-grained hope tone classification of code-mixed Tulu social media comments. Our method leverages multilingual transformer fine-tuning using the pre-trained *XLM-RoBERTa* model along with targeted preprocessing techniques designed for code-mixed text. Furthermore, we address class imbalance in the dataset using class-weighted loss and apply token-level augmentation to improve model robustness. In addition, we demonstrate the effectiveness of multilingual transformer models specifically for code-mixed Tulu, a highly underexplored and low-resource setting. These strategies help the system capture emotional and contextual information present in informal multilingual user-generated content.

Our major contributions can be summarized as follows:

- We present a multilingual transformer-based model for coarse-grained hope tone classification in code-mixed Tulu social media text.
- We introduce a preprocessing scheme that addresses the challenges of code-mixed data while preserving critical emotional cues such as emojis and colloquial expressions.

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 403–408
July 3, 2026 ©2026 Association for Computational Linguistics

- We incorporate class balancing and data augmentation techniques to improve macro-level performance.

2 Related Work

The major research in the field of Natural Language Processing (NLP) has been carried out to detect harmful content on the internet, including hate speech, abusive language, and offensive language. On the other hand, less research has been carried out on detecting constructive and positive content, including hope speech. Hope speech is defined as positive and encouraging content on the Internet. Chakravarthi proposed the concept of Multilingual Hope Speech Detection and the HopeEDI dataset, focusing on the promotion of Equality, Diversity, and Inclusion (EDI) using positive speech on the Internet in English and different Dravidian languages (Chakravarthi, 2022b).

Further research has also been conducted to extend the hope speech detection to social media environments, such as YouTube comments. For this purpose, deep learning techniques, such as CNN-based models, have been used for the classification of hope speech for different languages. The experiments have been conducted to prove the effectiveness of the neural network models for the detection of positive and supportive speech in social media environments (Chakravarthi, 2022a). The importance of the increasing need for analyzing positive speech, as well as negative speech, has also been highlighted.

Besides this, recent research has been carried out on the detection of hope speech in multilingual and code-mixed languages. A multilingual approach using transformer models such as XLM-RoBERTa and mBERT has been proposed for detecting hope speech in low-resource languages (Abdullah et al., 2025). The study showed that transformer-based architectures can effectively capture contextual information for hope speech classification.

Similarly, hope speech detection in code-mixed Roman Urdu has been explored by comparing traditional machine learning models with transformer-based techniques (Ahmad et al., 2025). The results indicate that transformer-based approaches outperform traditional machine learning methods when dealing with informal and orthographically inconsistent social media text.

Significant contributions have also been made in research on code-mixed Dravidian languages. A trilingual code-mixed Tulu dataset for sentiment analysis was introduced to better understand multilingual social media data (Hegde et al., 2022). In addition, a shared task on sentiment analysis for Tamil and Tulu code-mixed text highlighted the challenges of processing multilingual content written in Roman script (Hegde et al., 2023). Code-mixed text often contains multiple languages within the same sentence, making analysis more complex. Related research has also demonstrated the effectiveness of machine learning and deep learning approaches for sentiment analysis in Dravidian code-mixed languages, emphasizing the importance of contextual representation learning (Mishra et al., 2021).

3 Data Description

The dataset employed for this shared task consists of code-mixed Tulu social media comments collected from YouTube platforms (Thenmozhi et al., 2026). This is a natural dataset where Tulu is often mixed with English and Kannada, mostly written in the Roman script. Due to the informal nature of social media communication, the dataset contains spelling variations, irregular orthography, elongated characters, emojis, and grammatical errors. The label-wise distribution of the dataset is shown in Table 1.

The dataset is divided into three predefined splits: training, development, and test sets. Each instance contains a comment and a corresponding coarse-grained hope tone label. The task is formulated as a four-class classification problem with the labels *Encouraging*, *Discouraging*, *Uninvolved* and *Blended Tone*.

The class distribution is imbalanced, with the *Uninvolved* category containing more instances than the other classes. Therefore, Macro F1-score is used as the primary evaluation metric to assess model performance across all classes.

Label	Training Dataset	Development Dataset	Test Dataset
Blended Tone	895	191	172
Discouraging	711	153	149
Encouraging	1895	406	407
Uninvolved	2490	534	398
Total	**5991**	**1284**	**1126**

Table 1: Label-wise Distribution of Training, Development, and Test Data

4 Methodology

The proposed system uses a transformer model as a classification paradigm to detect coarse-grained

hope tone in code-mixed Tulu social media text. We have also provided the implementation of our system in a publicly available GitHub repository[1]. The overall workflow of the system includes preprocessing the noisy text, applying a light data augmentation technique, fine-tuning a multilingual transformer model, and addressing class imbalance to enhance macro-level performance. The system architecture is designed to effectively capture emotional content in informal and code-mixed text.

4.1 Task and Data Description

The shared task involves coarse-grained hope tone classification of social media comments written in code-mixed Tulu. Each comment is categorized into one of four classes: *Encouraging*, *Discouraging*, *Uninvolved*, and *Blended Tone*. The dataset consists of highly informal text collected from social media platforms such as YouTube, where code-mixing between Tulu, English, and Kannada is prevalent. The dataset contains spelling variations, Romanized script usage, emojis, and grammatical inconsistencies, making the classification task challenging. The organizers provided the dataset pre-split into training, development, and test sets. The final ranking of the systems was determined based on performance on the hidden test set.

4.2 Data Preprocessing

Because of the noisy and informal nature of code-mixed social media text, preprocessing was performed to improve text quality while preserving emotional information. URLs and unnecessary whitespace were removed. Repeated character sequences were normalized to reduce noise caused by elongated informal expressions. Unlike aggressive normalization techniques, emojis and other emotional indicators were retained as they carry important information useful for identifying hope tones. The text data was tokenized using the *XLM-RoBERTa tokenizer*. Padding and truncation were applied to ensure that all sequences had a uniform length of 128 tokens.

4.3 Data Augmentation

A random token deletion technique was employed to improve the robustness of the models trained on the smaller dataset. In this approach, tokens are randomly removed from the input text with a probability of 0.1, which results in the generation of data augmentation versions of the original input data.

4.4 Classification Model

For the classification of texts into hope tone, the pre-trained multilingual transformer model *XLM-RoBERTa-base* (Conneau et al., 2019) was fine-tuned. The model was selected due to its strong cross-lingual representation capability and effectiveness in handling code-mixed text classification tasks. A task-specific classification head was added on top of the pre-trained model. Fine-tuning was performed using the AdamW optimizer within the Hugging Face `Trainer` framework. The model was trained with a learning rate of 2×10^{-5}, a batch size of 16, and for 3 epochs.

4.5 Handling Class Imbalance

The dataset exhibits class imbalance across the four hope tone categories. To mitigate bias toward majority classes, class weights were computed from the training data distribution and incorporated into the cross-entropy loss function (Mao et al., 2023). This strategy improves macro-level performance, particularly the Macro F1-score, by enhancing prediction quality for minority classes.

4.6 Training Strategy

To ensure reproducibility and stable convergence, seed values were set in NumPy, PyTorch, and CUDA. The general flow of the model pipeline, starting from data preprocessing, to training and predicting, is described in Figure 1 below. The model was trained with the help of the Hugging Face `Trainer` class, evaluated on the validation set. The training was done on a GPU-enabled system.

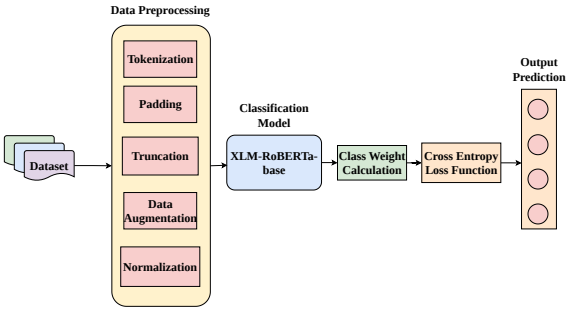


Figure 1: Proposed Transformer-based Hope Speech Detection Architecture

[1]The code repository is available at: GitHub Repository.

5 Results and Analysis

5.1 Leaderboard Performance

Team Oryu has achieved 3rd position in the task of coarse-grained hope tone classification in the shared task. For our submission, the system has achieved an Accuracy of 67%, Macro Precision of 64%, Macro Recall of 65%, and Macro F1-score of 63%. Considering the complexity of the task and the low-resource nature of the Tulu language, the proposed system has achieved a Macro F1-score higher than 60%, demonstrating the effectiveness of the proposed approach for code-mixed hope tone classification using a multilingual transformer model.

5.2 Performance Analysis

The multilingual transformer backbone model, XLM-RoBERTa, demonstrated strong performance in processing code-mixed Tulu–English–Kannada text by capturing cross-lingual dependencies and subtle emotional cues. The overall performance across evaluation metrics is shown in Table 2. The use of class weighting helped address class imbalance, while the lightweight data augmentation strategy improved generalization and overall macro-level performance.

Metric	Value (%)
Accuracy	67
Macro Precision	64
Macro Recall	65
Macro F1-score	63

Table 2: Performance Metrics of the Proposed System

5.3 Ablation Study

To evaluate the contribution of different components, we compare the baseline model with class balancing and the proposed system.

Model	Macro F1-score
XLM-RoBERTa-base	0.5554
XLM-RoBERTa-base + WeightedSampler	0.6220
Proposed Model	0.6276

Table 3: Comparison of baseline and proposed model performance

The results in the Table 3 show that class balancing significantly improves performance over the baseline, while the proposed model achieves the best performance.

5.4 Error Analysis & Discussion

Despite the fact that the predictive capabilities of the model with regard to accuracy and F1-scores were impressive, several challenging cases remained, as reflected in the confusion matrix in Figure 2. One of them was *Blended Tone*, which, due to the presence of emotions in one statement, led to either *Encouraging* or *Discouraging* classification. In addition, there was the possibility of misclassification of Discouraging as either Blended Tone or Encouraging. For instance, comments where encouraging messages were combined with sarcastic remarks caused errors in classification, which means that the model could not cope with such nuanced changes of emotions in the text. Also, lack of context, semantics of code-mixed language and alternative spellings in Roman alphabets led to errors. Some Uninvolved comments could even be confused with Blended Tone. The results demonstrate the effectiveness of multilingual transformer models for the detection of emotional signals in code-mixing languages. While our system is outperformed by the best-performing model (Macro-F1: 58%), it achieves comparable performance, despite the room left for improvement in terms of the difference in performance metrics. More advanced approaches to increasing the accuracy of emotion recognition include applying focal loss and ensemble learning with more extensive transformer models. It is also possible to increase the performance of the model through improved classification of ambiguous categories like the *Blended Tone*. Further research might involve exploring other forms of data augmentation apart from the existing one.

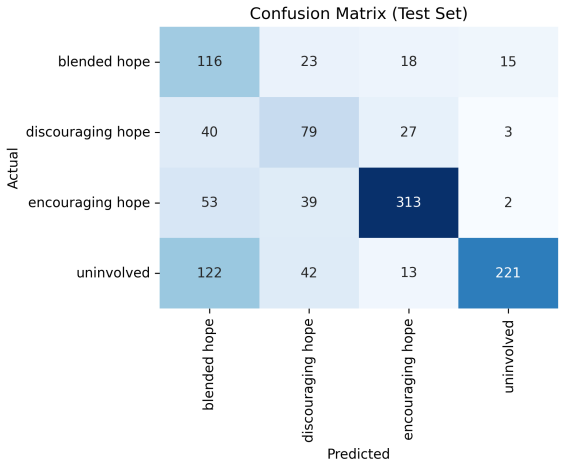


Figure 2: Confusion Matrix on Labeled Test Set

6 Conclusion

In this paper, we describe the system developed by Team Oryu for the shared task on Hope Speech Detection in Code-Mixed Tulu. The system employed multilingual transformer fine-tuning, preprocessing, handling of class imbalance, and data augmentation techniques. The system achieved a Macro F1-score of 63% on the test data, which enabled us to secure the 3rd place. The results show the promise of multilingual pre-trained models in detecting emotional cues in code-mixed social media text, and also reveal the difficulties of tone ambiguity. The potential avenues for further research will be to incorporate ensemble methods, data augmentation, focal loss, and larger multilingual pretrained models to enhance the performance of the systems.

Limitations

Despite the achievement of competitive performance, the proposed system has some limitations. First, the dataset employed in the current research has limited size and is in a low-resource language setting, which limits the ability of the model to learn complex linguistic patterns. Second, the employed dataset has highly informal code-mixed text, including spelling variants, inconsistent Romanization, and mixing of languages such as Tulu, English, and Kannada, which makes the interpretation of the text challenging for the model. Although the multilingual transformer-based model such as XLM-RoBERTa can effectively interpret the context of the given sentence, it is trained on large multilingual datasets that may not effectively cover the linguistic patterns of the Tulu language. Moreover, the employed dataset has class imbalance, which may affect the predictions of the model towards the classes that have the most occurrences despite the application of class weighting to the model. Lastly, because of the computational and time limitations of the current research, the model was not able to perform extensive experiments using other transformer-based models such as large transformer variants. In future research, we plan to improve the model further by utilizing a larger multilingual transformer model and/or ensemble model to improve the classification performance. Also, a larger dataset and better data preprocessing techniques for code-mixed text might improve the model's generalization performance.

Ethical Considerations

The work conforms to ethical guidelines for responsible research in the field of Natural Language Processing. The data set employed in the research for this paper is publicly available YouTube comments provided by the organizers of the shared task, and no efforts were made to retrieve any personal information of the users of the YouTube platform. Hence, the work maintains the privacy of the users and does not handle any sensitive personal information of the users of the YouTube platform. Since social media data may have bias in terms of language or culture, it should be noted that bias may be present in the predictions made by the model. To avoid any bias in the predictions made by the model, balanced measures were used to evaluate the model's performance, such as Macro F1-Score. The proposed system aims to be used for research to facilitate constructive online discourse and should not be used for moderation. We also recognize that the data is from a low-resource language environment and may not adequately represent the linguistic spectrum of the Tulu-speaking people. Future work could include a larger and more representative data set to promote fairness and inclusiveness.

References

Ahmed Abdullah, Sana Fatima, and Haroon Mahmood. 2025. Ghalib: A multilingual framework for hope speech detection in low-resource languages. *arXiv preprint arXiv:2512.22705*.

Muhammad Ahmad, Muhammad Waqas, Ameer Hamza, Ildar Batyrshin, and Grigori Sidorov. 2025. Hope speech detection in code-mixed roman urdu tweets: A positive turn in natural language processing. *arXiv preprint arXiv:2506.21583*.

Bharathi Raja Chakravarthi. 2022a. Hope speech detection in youtube comments. *Social Network Analysis and Mining*, 12(1):75.

Bharathi Raja Chakravarthi. 2022b. Multilingual hope speech detection in english and dravidian languages. *International Journal of Data Science and Analytics*, 14(4):389–406.

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.

Asha Hegde, Mudoor Devadas Anusha, Sharal Coelho, Hosahalli Lakshmaiah Shashirekha, and Bharathi Raja Chakravarthi. 2022. Corpus creation

for sentiment analysis in code-mixed tulu text. In *Proceedings of the 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages*, pages 33–40.

Asha Hegde, Bharathi Raja Chakravarthi, Hosahalli Lakshmaiah Shashirekha, Rahul Ponnusamy, Subalalitha Cn, Lavanya SK, Martha Karunakar, Shreya Shreeram, Sarah Aymen, et al. 2023. Findings of the shared task on sentiment analysis in tamil and tulu code-mixed text. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, pages 64–71.

Anqi Mao, Mehryar Mohri, and Yutao Zhong. 2023. Cross-entropy loss functions: Theoretical analysis and applications. *Preprint*, arXiv:2304.07288.

Ankit Kumar Mishra, Sunil Saumya, and Abhinav Kumar. 2021. Sentiment analysis of dravidian-codemix language. In *FIRE (Working Notes)*, pages 1011–1019.

Durairaj Thenmozhi, Rathnakar Shetty P, Parameshwar R. Hegde, Anusha M D, Raksha Adyanthaya, Mohammed Fadhel Aljunid, Prasanna Kumar Kumaresan, and Bharathi Raja Chakravarthi. 2026. Findings of the Shared Task on Hope Speech Detection in Tulu. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Team_One@DravidianLangTech 2026: A Gated Multimodal Architecture for Multi-Level Stance and Target Detection in Malayalam Political Memes

Ashmi S N[1], **Nimisha M Iyer**[1], **Balasubramanian Palani**[1]
Jobin Jose[1], **Siranjeevi Rajamanickam**[2]
[1]Indian Institute of Information Technology Kottayam, Kerala, India
[2]Govt. Polytechnic College-Thuvakudi, Tiruchirappalli, Tamil Nadu, India
ashmi23bcs196@iiitkottayam.ac.in, nimisha23bcd4@iiitkottayam.ac.in
pbala@iiitkottayam.ac.in, jobin@iiitkottayam.ac.in, rajasiranjeevi@gmail.com

Abstract

Stance and target detection in multimodal political memes presents notable challenges in low-resource and highly imbalanced settings. This task is based on the Malayalam dataset from the DravidianLangTech 2026 Shared Task (500 samples with a 95.4:4.6 stance imbalance). The primary challenges stem from linguistic variability and visually complex meme formats, which hinder accurate text extraction and effective multimodal alignment. A lightweight yet high-performing multimodal framework is proposed that integrates bilingual OCR, a Vision Transformer (ViT), and IndicBERT to learn complementary visual and textual representations. A gated fusion mechanism effectively combines multimodal features, while asymmetric loss weighting and post-training threshold optimization address extreme class imbalance. The methodology achieves a Weighted F1-score of 0.9535 for stance detection and 0.5283 for target identification, demonstrating strong robustness and generalization under realistic multimodal constraints. The training pipeline is publicly available for reproducibility.[1]

1 Introduction

Political internet memes generally depend on the intricate combination of visual signals and accompanying text that are often laden with local sarcasm, code-mixed expressions, and slang. The DravidianLangTech 2026 Shared Task (Rajiakodi et al., 2026) challenges systems to automate this understanding across two hierarchical levels: Level 1 (Stance Detection), a binary classification of Support vs. Oppose; and Level 2 (Target Identification), a 5-class fine-grained classification identifying the specific target.

This task is severely hampered by an ultra-low-resource training environment and an extreme lack of minority class examples. Standard multilingual models and simplistic late-fusion techniques frequently fail to capture the semantic interplay between modalities under such scarcity. In this paper, a comprehensive pipeline tailored for Malayalam is detailed. By combining bilingual Tesseract Optical Character Recogonition (OCR) (Smith, 2007), along with the Vision Transformer (ViT) architecture (Dosovitskiy et al., 2021), leveraging IndicBERT (Kakwani et al., 2020) for multilingual contextual encoding, and introduce a mathematically robust Gated Attention fusion module inspired by attention gating mechanisms (Ilse et al., 2018). This architecture performs simultaneous multi-level classification, effectively isolating rare minority classes without sacrificing majority-class precision.

2 Related Work

Multimodal Meme Analysis While models like CLIP (Radford et al., 2021) revolutionized multimodal alignment, their English-centric training limits regional applicability. Processing localized memes requires robust OCR. Because OCR extractions from stylized memes are inherently noisy, specialized fusion techniques are required to dynamically process garbled text based on visual context (Suryawanshi et al., 2020). Traditional architectures relied on CNNs (He et al., 2016), but Vision Transformers (Dosovitskiy et al., 2021) now allow networks to capture global spatial relationships via self-attention from the first layer.

Dravidian NLP & Encoders Code-mixing on social media complicates standard NLP pipelines (Chakravarthi et al., 2020). Three pre-trained architectures were benchmarked: XLM-RoBERTa ($\approx$ 270M parameters) (Conneau et al., 2020), MuRIL ($\approx$ 236M parameters) (Khanuja et al., 2021), and IndicBERT ($\approx$ 33M parameters) (Kakwani et al., 2020). Built on ALBERT (Lan et al., 2020), In-

[1]Code available at: https://github.com/ashmisn/dravidlangtech

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 409–413
July 3, 2026 ©2026 Association for Computational Linguistics

dicBERT employs cross-layer parameter sharing that acts as a structural regularizer against overfitting, making it well-suited for agglutinative Malayalam text.

3 Proposed Methodology

3.1 OCR and Data Augmentation

Text extracted via bilingual Tesseract OCR (`mal+eng`) is regex-filtered to retain only Malayalam Unicode and English alphanumerics, minimizing out-of-vocabulary tokens to yield sequence S. To combat data scarcity, standard image augmentations (flipping, rotation, brightness) are applied. The complete architecture is illustrated in Figure 1.

3.2 Textual Encoding via IndicBERT

Sequence S is encoded using IndicBERT. By utilizing cross-layer parameter sharing, it reduces its footprint to $\approx$ 33M parameters, acting as a natural regularizer against overfitting. It employs standard multi-head self-attention, projecting inputs into Query (Q), Key (K), and Value (V) matrices:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

where d_k is the key dimensionality. This yields a dense contextualized representation $H_{text} \in \mathbb{R}^{768}$. For multimodal fusion, H_{text} is projected into a shared latent space ($d_{proj} = 512$) via linear mapping, Layer Normalization (LN), and ReLU:

$$z_{text} = \text{ReLU}(\text{LN}(W_t H_{text} + b_t)) \quad (2)$$

where $W_t \in \mathbb{R}^{512\times768}$ and $b_t \in \mathbb{R}^{512}$ are learnable projection parameters.

3.3 Visual Encoding via ViT

Image I is processed by a pre-trained Vision Transformer (ViT-Base). ViT divides the image into 16×16 patches and applies the identical self-attention mechanism (Equation 1). This captures global spatial dependencies directly from the `[CLS]` token, outputting visual representation $H_{vis} \in \mathbb{R}^{768}$. Similar to the textual pathway, H_{vis} is identically projected into the shared latent space:

$$z_{vis} = \text{ReLU}(\text{LN}(W_v H_{vis} + b_v)) \quad (3)$$

where $W_v \in \mathbb{R}^{512\times768}$ and $b_v \in \mathbb{R}^{512}$ are the visual projection parameters.

3.4 Gated Attention Fusion

Simple concatenation propagates OCR noise directly into the visual context (Tsai et al., 2019). To enable adaptive filtering, our architecture leverages the structurally aligned latent embeddings z_{vis} and z_{text}. A Sigmoid (σ) gate evaluates their concatenated features to learn a dimension-wise weight vector $\alpha \in (0, 1)$:

$$\alpha = \sigma(W_g[z_{vis} \oplus z_{text}] + b_g) \quad (4)$$

The final joint representation dynamically balances the modalities via element-wise multiplication ($\odot$):

$$F = \alpha \odot z_{text} + (1 - \alpha) \odot z_{vis} \quad (5)$$

If unstructured OCR noise is detected, the network learns to drive $\alpha \to 0$, filtering the corrupted text and forcing reliance on robust visual cues.

3.5 Multi-Task and Asymmetric Learning

The fused vector F passes through a shared Fully Connected block with Dropout ($p = 0.4$) to yield a joint representation $R \in \mathbb{R}^{256}$. This shared latent space facilitates multi-task learning by transferring beneficial inductive biases between the stance and target classification heads. To address extreme class imbalance without relying on generative upsampling (which can distort compact embedding spaces), class-specific asymmetric weights are applied to the Cross-Entropy loss:

$$\mathcal{L} = -\sum_{i=1}^{C} \lambda_i y_i \log(P(\hat{y}_i|I, S)) \quad (6)$$

Setting $\lambda_{minority} = 3.0$ and $\lambda_{majority} = 1.0$ artificially inflates the gradient magnitude for minority errors, heavily penalizing misclassifications on rare classes.

3.6 Post-Training Threshold Optimization

Standard binary classifiers default to a static decision boundary of $\tau = 0.50$. Under a severe 95.4:4.6 skew, continuous probability outputs inherently bias toward the prior distribution, often leading to majority-class collapse. Instead of modifying the data distribution, the decision logic is optimized. Post-training, validation thresholds $\tau \in [0.05, 0.50]$ are systematically swept to identify the optimal boundary τ^* that maximizes the primary evaluation metric (Macro F1):

$$\tau^* = \arg\max_{\tau} F1_{Macro}(\tau) \quad \text{s.t.} \quad TP > 0 \quad (7)$$

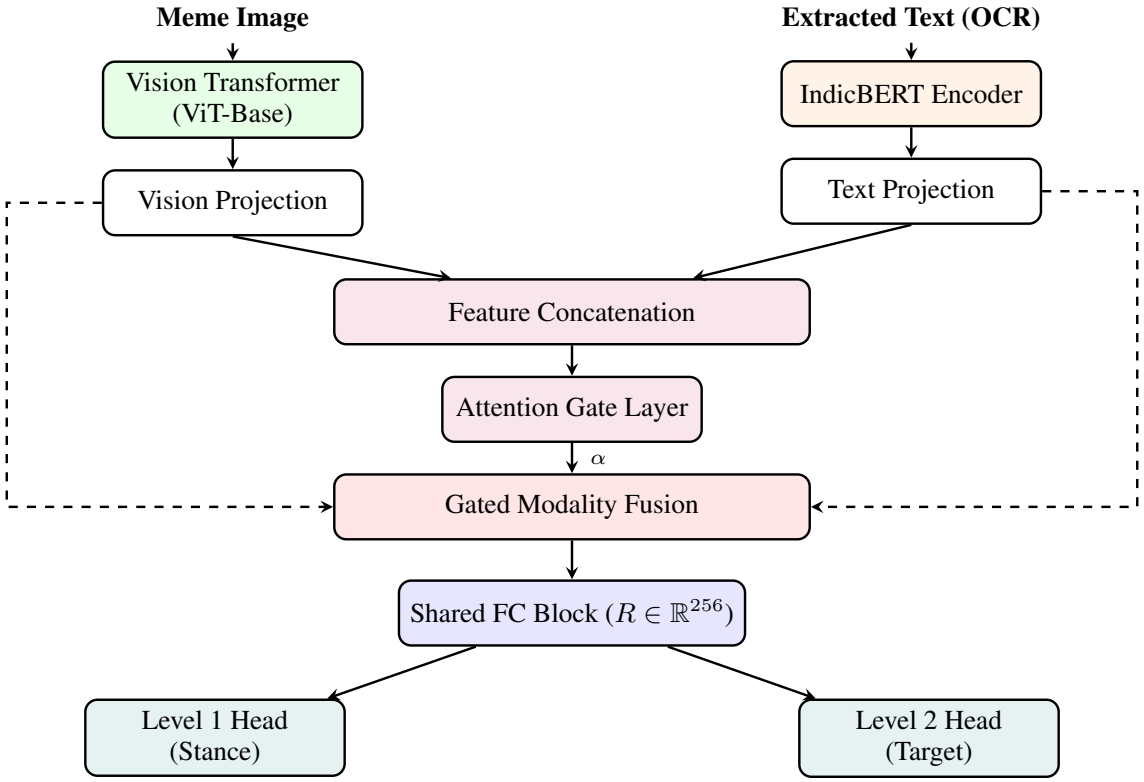


Figure 1: Gated Multimodal Architecture. The dynamic attention gate calculates optimal fusion weights before passing the unified representation to multi-level classification heads.

This calibration explicitly aligns the model's predictive boundary with the task's minority-sensitive objectives.

4 Experimental Setting

4.1 Implementation Setup

Experiments were conducted on an NVIDIA Tesla T4 GPU using PyTorch and HuggingFace Transformers. Training was performed for 12 epochs using the AdamW optimizer. A batch size of 8 was used due to memory constraints, which also introduced mild stochastic regularization.

To prevent catastrophic forgetting, a conservative learning rate of $1e^{-5}$ was applied to the pretrained ViT and IndicBERT encoders, while a larger rate of $1e^{-4}$ was assigned to the randomly initialized fusion layers to enable faster convergence. A decoupled weight decay of $1e^{-2}$ was used to further control overfitting.

4.2 Dataset Description

The dataset distribution is presented in Table 1.

4.3 Performance Metrics

To evaluate performance under severe class imbalance, Accuracy ($\frac{TP+TN}{TP+TN+FP+FN}$) is used, where

Table 1: Class Distribution (Train vs. Test Set)

Level 1 (Stance)	**Train (n=500)**	**Test (n=100)**
TROLL / OPPOSE	477 (95.4%)	96 (96.0%)
Support/Praise	23 (4.6%)	4 (4.0%)
Level 2 (Target)	**Train (n=500)**	**Test (n=100)**
Against individual person	315 (63.0%)	50 (50.0%)
Against party	110 (22.0%)	31 (31.0%)
Intersection	53 (10.6%)	15 (15.0%)
Support for individual	12 (2.4%)	4 (4.0%)
Support for party	10 (2.0%)	0 (0.0%)

TP, TN, FP, and FN denote True Positives, True Negatives, False Positives, and False Negatives, respectively. Macro F1 ($\frac{1}{C}\sum_{i=1}^{C} F1_i$) is prioritized, as it computes the unweighted mean of the C class-wise F1-scores and assigns equal importance to minority classes. Additionally, Weighted F1 ($\sum_{i=1}^{C} w_i F1_i$) scales the scores by the true proportion (w_i) of samples in each class.

5 Results and Discussion

5.1 Level 1 (Binary Classification): Stance Detection Results

Table 2 presents the ablation results. Although MuRIL + ViT achieves high accuracy, its low

Table 2: Performance of models on Binary Classification (Level 1)

Model	Macro F1	Weighted F1	Accuracy
XLM-R + ViT	0.5105	0.9210	93.5%
MuRIL + ViT	0.4898	0.9404	96.0%
IndicBERT + ResNet	0.5400	0.8700	86.2%
IndicBERT + ViT	**0.6564**	**0.9535**	**96.0%**

Table 3: Performance of models on Multiclass Classification (Level 2)

Model	Precision	Recall	F1-score
XLM-R + ViT	0.5010	0.4680	0.4805
MuRIL + ViT	0.4920	0.4520	0.4708
IndicBERT + ResNet	0.5180	0.4890	0.5012
IndicBERT + ViT	**0.5471**	**0.5500**	**0.5283**

Table 4: Class-wise Performance of IndicBERT+ViT (Level 2)

Target Class	Precision	Recall	F1-score
Against indv. person	0.5441	0.7400	0.6271
Against party	0.5417	0.4194	0.4727
Intersection	**0.7143**	0.3333	0.4545
Support for individual	0.0000	0.0000	0.0000
Weighted Average	**0.5471**	**0.5500**	**0.5283**

Macro F1 indicates inadequate minority class detection, a byproduct of the extreme data imbalance. IndicBERT + ResNet improves minority sensitivity slightly but suffers a drop in accuracy, suggesting weaker visual feature extraction.

The proposed IndicBERT + ViT model achieves the best balance. Reinforced by the asymmetric loss weighting and threshold calibration ($\tau^* = 0.10$), the model achieves a Macro F1-score of 0.6564. It successfully detects scarce "Support" memes without compromising the 96.0% majority class stability.

5.2 Level 2 (Multiclass classification): Target Identification Results

Table 3 compares the overall multiclass classification performance across different architectural combinations. The proposed IndicBERT + ViT fusion achieves the highest precision, recall, and F1-score, demonstrating the effectiveness of combining strong linguistic representations with robust visual feature extraction for fine-grained target identification.

The detailed breakdown in Table 4 reveals class-specific behavior. The model achieves its highest F1-score for the dominant "Against individual person" category, indicating stable learning for majority targets. The "Intersection" class records the highest precision (0.7143), suggesting that the gated fusion mechanism effectively captures overlapping target semantics. However, its comparatively lower recall reflects limited exposure during training. The "Support for individual" category records zero performance. This outcome primarily stems from extreme data sparsity, with only a small number of training instances and the absence of sufficient representative samples in the evaluation split. Such scarcity prevents the formation of reliable and generalizable decision boundaries.

Heavily stylized and code-mixed typography significantly degrades standard OCR extraction quality. Naive late-fusion approaches tend to propagate this textual noise into the final representation. In contrast, the gated attention mechanism adaptively filters unreliable signals. When high-entropy OCR outputs are detected, the sigmoid gate suppresses the corrupted textual modality and shifts reliance toward visual embeddings (z_{vis}), thereby preserving classification robustness despite severe OCR degradation.

6 Conclusion and Future Work

Processing code-mixed slang in ultra low-resource meme datasets remains a challenging task due to noisy text extraction and severe class imbalance. To address these challenges, this paper introduces a robust, gated multimodal architecture specifically tailored for multi-level stance and target detection in Malayalam political memes. Unlike existing approaches that predominantly cater to high-resource languages or rely on standard feature concatenation, our framework uniquely integrates advanced multimodal gating with imbalance-aware training. This ensures the model effectively captures the nuanced synergy between stylized code-mixed text and visual cues, providing a significant improvement over traditional baselines in low-resource settings.

Future work will focus on improving OCR quality for code-mixed and stylized meme text, particularly by fine-tuning OCR models on Malayalam-English slang and low-resolution meme images. Additionally, expanding annotated data for minority classes by collecting more diverse and representative samples will be crucial to strengthening model robustness and improving its applicability in real-world content moderation scenarios.

7 Limitations

OCR extraction remains a major limitation in Malayalam political memes due to highly stylized fonts, low-resolution images, and heavy Malayalam-English code-mixing. Although the proposed gated fusion mechanism reduces the impact of noisy OCR outputs, overall performance still depends significantly on the quality and completeness of extracted text.

8 Ethical Considerations

Automated moderation of subjective Malayalam sarcasm risks inadvertently flagging benign political discourse. Hence, this model is intended strictly for human-in-the-loop moderation, not autonomous censorship. Additionally, AI tools were used solely for proofreading; all methodologies and analyses remain entirely human-authored.

References

B. R. Chakravarthi, V. Muralidaran, R. Priyadharshini, and J. P. McCrae. 2020. Corpus creation for sentiment analysis in code-mixed tamil-english text. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages*, pages 202–210.

A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8440–8451.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*.

K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778.

M. Ilse, J. M. Tomczak, and M. Welling. 2018. Attention-based deep multiple instance learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 2127–2136. PMLR.

D. Kakwani, A. Kunchukuttan, S. Golla, G. N.C., P. Bhattacharyya, M. M. Khapra, and P. Kumar. 2020. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961. Association for Computational Linguistics.

S. Khanuja, D. Dikshit, P. Talukdar, M. Hasan, A. Agrawal, and C. Carvallo. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, and D. Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. *NeurIPS*.

Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations (ICLR)*.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763.

S. Rajiakodi, S. P. M. Chinnan, B. Premjith, C. Subalalitha, P. Rahul, K. Anshid, S. Bhuvaneshwari, V. Jananayagan, N. Ragavan, P. Santhini, and B. R. Chakravarthi. 2026. Shared task on multi-level political meme classification for dravidian languages. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*.

R. Smith. 2007. An overview of the tesseract ocr engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE.

S. Suryawanshi, B. R. Chakravarthi, P. Verma, M. Arcan, and P. Buitelaar. 2020. Multimodal meme dataset (multioff) for identifying offensive content in image and text. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 14–22.

Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov. 2019. Multimodal routing: Improving local and global interpretability of multimodal language analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1823–1833.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems (NeurIPS)*, 30.

Trailblazer@DravidianLangTech 2026: A Comparative Study of TF-IDF SVM and XLM-RoBERTa for Political Multiclass Text Classification.

Shanthi Murugan , Anbuaruvi R , Anuradha C
Department of Artificial Intelligence and Data Science,
R.M.K. Engineering College
Kavaraipettai,Chennai, India
msi.ad@rmkec.ac.in, 230204.ad@rmkec.ac.in, 230042.ad@rmkec.ac.in

Abstract

The rapid growth of social media networks faces challenges in the classification of multilingual and code-mixed data. A task is shared by Political Multiclass Sentiment Analysis of Tamil X (Twitter) -DravidianLangTech@ACL 2026 to classify the political text.For the above task, we proposed solutions to compare a traditional machine learning and the transformer based model. First we developed a Baseline traditional Support vector Machine model using the TF-IDF features. To provide a stronger Indic-language baseline we consider the IndicBERT, a transformer model specifically designed for Indian Languages.IndicBERT improves contextual understanding of Tamil-English code-mixed political text . To capture the deeper information from the text we developed an XLM-RoBERTa model where we used minimal pre-processing technique. The Result shows us that the transformer-based performs well compared to the traditional baseline model with the macro F1 score of 0.3738. The Study highlights the importance of robust multi-class social media political text classification.

1 Introduction

The rapid expansion of social media platforms has transformed the way individuals communicate, share the opinions. Millions of posts are generated daily. The number of people who use social media is projected to grow exponentially, reaching nearly 1.2 billion users by 2029. In recent days, code-mixed text is used everywhere mainly in social media post. The task of sentiment analysis became difficult due to the usage of informal grammar, emojis, abbreviations, sarcasm, mixed language. The shared task on the Political Multiclass Sentiment Analysis of Tamil X (Twitter) - DravidianLangTech@ACL-2026 - aimed to address this challenge by identifying the types of political sentiments such as Substantiated, Sarcastic, Opinionated, Positive, Negative, Neutral, and None of the above. In Traditional machine learning approaches, faces struggle to capture the meaning of the text. Advancements in transformer-based models have significantly improved contextual language understanding.

In our participation, we focus on addressing the challenges of political sentiment classification through three primary contributions:

• We implemented three pre-processing strategies to bring the meaningful text understand by the model.

• We implement a Comparative framework between traditional ML model and the transformer based models

•We evaluated models using macro-F1 score to ensure balanced assessment across all sentiment categories rather than favoring majority classes.

Our code, developed for this shared task can be accessed here https://github.com/Anbuaruvi/Trailblazer.git

2 Related Works

The complexity of understanding and categorizing political sentiments has driven extensive research employing various languages, datasets, and methodologies. Research has been done to advance sentiment analysis in under-resourced code-mixed languages(Kumar et al., 2024) . Different machine learning models and Deep Learning models have been employed to classify sentiment of text(Barua et al., 2025). (Gandhar et al., 2024)conducted a comparative analysis of machine learning approaches such as Support Vector Machines (SVM) for sentiment analysis in Indian multilingual social media contexts using Twitter data. Their work highlights the challenges posed by mixed-language and code-mixed text data, and shows that SVM remains effective but limited for nuanced sentiment detection.Due to the inefficiency of machine learning models in extracting contex-

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 414–419
July 3, 2026 ©2026 Association for Computational Linguistics

tual meanings, their works lack the ability to fully capture nuanced expressions and complex sentiment patterns.(Khurana and Prabhu, 2023)experiments on labeled Twitter data using transformer models such as BERT,ALBERT,RoBERTa and DeBERTa ,showing that fine-tuned transformer models significantly performs better than the classical machine learning in capturing sentiment in social media text.Another recent research in sentiment work includes multilingual sentiment analysis using the transformer model,this models support multiple Indian Languages(Kalla, 2025).Multilingual transformers have been explored for multiclass sentiment analysis in code-mixed data, effectively capturing contextual nuances in low-resource languages (Nazir et al., 2025). Collectively, these traditional and hybrid studies highlight the evolution of sentiment analysis research from feature-based and lexicon methods to more sophisticated context-aware models.

3 Task and Dataset Description

The Shared task focuses on the analysis of political multiclass sentiment of Tamil X (Twitter) post comments in Tamil language,which are annotated with seven categories .The main objective is to classify the sentiment from the given text.The Seven labels are as follows:

- **Substantiated** – Supported by facts, statistics, references, or specific details.
- **Sarcastic** – Sentiment based on sarcasm, mockery, or irony.
- **Opinionated** – Sentiment based on strong personal views or individual perspectives.
- **Positive** – Sentiment expressing support, approval, or appreciation.
- **Negative** – Sentiment based on criticism or dissatisfaction.
- **Neutral** – Sentiment based on general political information or factual updates without expressing opinions, emotions, or bias.
- **None of the above** – Sentiment that does not fit into any of the specified categories.

The distribution of the political sentiment classes across the training, development, test (Given), and Pred (Predicted) datasets is shown in Table 4. The training dataset exhibits a noticeable class imbalance, with the "Opinionated" category having the highest representation, significantly outnumbering other classes. In contrast, categories like "None" and "Substantiated" have relatively fewer samples, which is illustrated in Figure 3.

3.1 Tables and Visualization

Table 1: Label distribution across Train, Dev, Test (Given) and Predicted datasets.

Label	Train	Dev	Test	Pred
Substantiated	412	52	51	60
Sarcastic	790	115	106	102
Opinionated	1361	153	171	118
Positive	575	69	75	132
Negative	406	51	46	73
Neutral	637	84	70	36
None	171	20	25	23

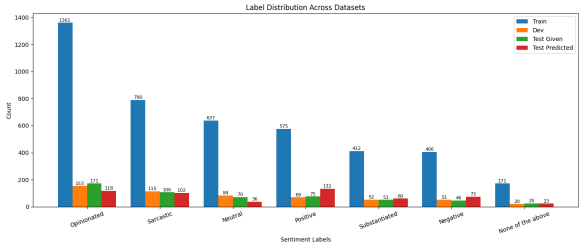


Figure 1: Distribution of sentiment labels across datasets.

4 Methodology

The Methods and Strategies used to classify the sentiment lables are detailed in this section.We implemented three different approaches:

1. A Baseline ML approach using TF-IDF with Support Vector Machine(SVM).(Ardiansyah et al., 2024)
2. An Indic-language transformer baseline using IndicBERT. (Kakwani et al., 2020)
3. A transformer-based approach using XLM-RoBERTa. (Liu et al., 2019)

4.1 Pre-Processing

4.1.1 Baseline SVM

The Traditional ML model requires careful preprocessing. The following are the applied preprocessing steps. Removal of URLs and Mentions like username, hastags , emojis , special symbols. This does not provide any useful meaning for sentiment analysis. Converting all the text to lowercase also removes the extra spaces.

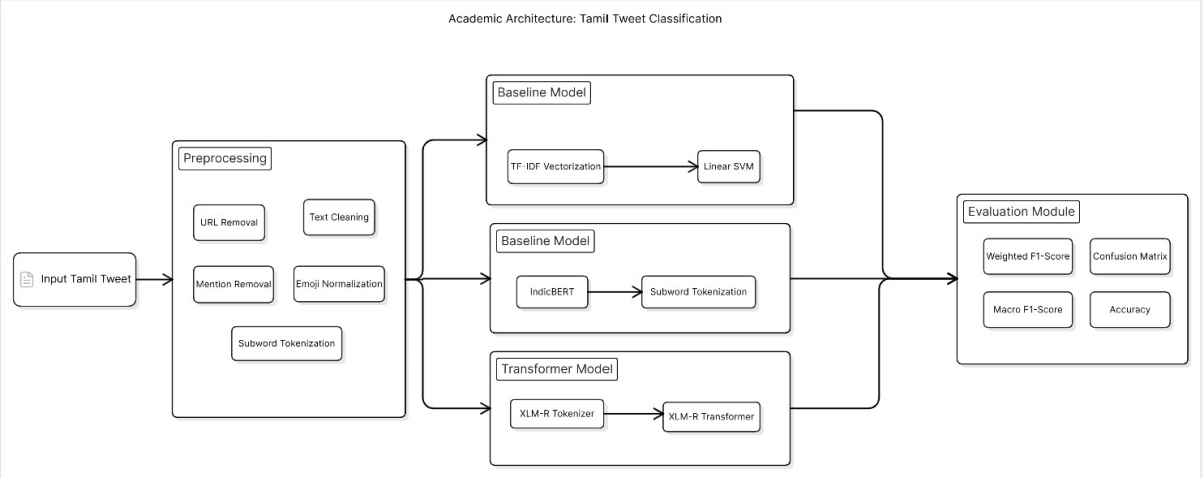


Figure 2: An Abstract view of Methodology

4.1.2 IndicBERT

The preprocessing steps applied for the IndicBERT include removal of URLs ,User mentions and unnecessary whitespace normalization. Unicode normalization was performed performed to maintain consistent Tamil text representation. Hashtags, punctuation, and casing information retained to preserve contextual and semantic meaning. The text was tokenized using subword tokenization for contextual embedding.

4.1.3 XLM-RoBERTa

Transformer-based models such as XLM-RoBERTa-base use subword tokenization and contextual embeddings.The steps performed are removal of URLs,user mentions, hastag symbol, whitespace normalization. No lowercasing or aggressive symbols removal is performed. The Minimal processing preserves semantics richness.

4.2 Modelling

4.2.1 Baseline SVM

The preprocessed text is converted into numerical feature vector using the Term Frequency-Inverse Document Frequency(TF-IDF). The n-gram range used here is unigram and bigrams using the TF-IDF features trained the Linear Support vector Machine.

The SVM Model is trained using the following to overcome the class imbalance:

1.Regularization parameter is set to C=1.0.

2.To handle class imbalance, the SVM was configured with class weight = "balanced", which assigns weights inversely proportional to class frequencies during training.

Table 2: IndicBERT Training Hyperparameters

Hyperparameter	Value
Learning Rate	2×10^{-5}
Batch Size	16
Epochs	5
Maximum Sequence Length	128
Warmup Ratio	0.1
Weight Decay	0.01

4.2.2 IndicBERT

The IndicBERT model was fine-tuned using the preprocessed Tamil X dataset for seven-class sentiment classification. The input text was tokenized using the IndicBERT tokenizer with padding and truncation applied to maximum sequence length of 128 tokens. Weighted cross-entropy loss and AdamW optimizer were used to address class imbalance and optimize training.

Table 3: XLM-RoBERTa Training Hyperparameters

Hyperparameter	Value
Learning Rate	2×10^{-5}
Batch Size	16
Epochs	6
Warmup Ratio	0.1
Weight Decay	0.01

4.3 XLM-RoBERTa

The transformer-based model used here is XLM-RoBERTa-base a mulitlingual transformer which is pre-trained on large languages .It consist of a token embedding layer, positional encodings, multiple

staked encoder blocks and a feed-forward networks and a final classification layer. The text input is processed using the XLM-RoBERTa tokenizer with a subword tokenization, truncation and padding to a maximum sequence of 128 tokens and attention to ensure uniform batch processing.

To Overcome class imbalance, class weights are computed from training data are incorporated into loss function and the model is optimized using the cross-entropy.

5 Results and Discussion

In this section, let us compare how the baseline SVM model, IndicBERT model and Transformer based XLM-RoBERTa perform on the given data. The effectiveness of the model is primarily assessed based on the F1 Score.The baseline model achieved the accuracy of 29.96 percentage with a Macro-F1 score of 0.2994 and the Weighted-F1 score of 0.2988.The low Macro-F1 score indicates that the model id difficulty in handling the imbalanced multi-class labels. Overall, the baseline model struggled to understand the semantic structure. IndicBERT perform better than traditional TF-IDF-based machine learning models because of its ability to capture contextual information and semantic relationships in Indic languages. The IndicBERT model achieved an accuracy of 0.3210 with a Macro-F1 score of 0.3345, outperforming the baseline TF-IDF + SVM model by effectively capturing contextual and semantic information present in Tamil-English code-mixed political text.

The transformer-based XLM-RoBERTa model performs better than the Baseline SVM.The model is trained using 6 epoches where result is The Accuracy 33.63 percentage,Macro-F1 score has 0.3640, Weighted-F1 score is 0.3395. The training log shows an improvement in each epoch from Epoch 1 to Epoch 6. Due to the stable increase overfitting did not occur. The transformer-based multilingual model XLM-RoBERTa learn contextual representation through self-attention mechanism,enabling better understanding of semantic relationships across multilingual text which makes transformer model work well compared to the baseline model. The Baseline SVM Model shows different performance across the class labels. In Class Label wise, the "None of the above", "Opinionated", "Sarcastic" shows higher Macro-F1 Score. Followed by The labels "Neutral", "Negative", "Positive" show the lower Macro-F1 score. The dataset contains significant class imbalance particularly with the "Opinionated" category dominating the training distribution while "Substantiated" and "None of the above" contains fever samples.

To Overcome this class imbalance, weighted cross-entropy loss is applied in the transformer model, also Macro-F1 is additionally added to evaluate the balanced performance across all the class labels.

From the baseline SVM Model to the Transformer-based model, the Macro-F1 score has increased approximately 6.5 percentage.

Table 4: Accuracy and Macro-F1 score for three different Models

Model	Accuracy	Macro-F1 score
SVM(Baseline)	0.2996	0.2994
IndicBERT	0.3210	0.3345
XLM-RoBERTa	0.3363	0.3640

Figure 3 represents the confusion matrix of the fine-tuned XLM-RoBERT model. The matrix provides a class label wise correct and incorrect predictions across all the seven labels. The highest number of correct prediction appears in the class label called **Opinionated** class which indicates that the transformer captures strong subjective expression. The class label **Sarcastic** shows robust performance with **56** correct predictions. The class label **None of the above** also classified the text with high precision. A significant number of Negative and Neutral labels are misclassified as Opinionated or Positive. The confusion matrix reveals that the transformer model successfully captures contextual relationships.

5.1 Error Analysis

Despite the improvement achieved by XLM-RoBERTa several challenges also remain.

- 1. Misclassification occurs between Negative, Neutral, Opinionated, and Positive sentiment classes. Class imbalance in the dataset causes the model to become biased toward majority classes such as Opinionated, leading to incorrect predictions for minority classes.
- 2. Code-mixed Tamil-English text creates difficulty in understanding contextual meaning and semantic relationships. Informal spelling variations and noisy social media text reduce the discriminative capability of the model.

- 3. Political social media discussions often contain emotions, sarcasm, ambiguous wording, and rhetorical expressions.

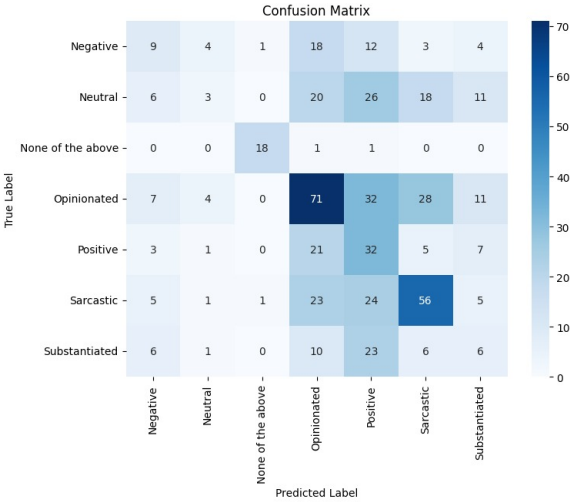


Figure 3: Confusion Matrix for XLM-RoBERTa

6 Conclusion

In this work, we proposed three different approaches for multiclass political sentiment classification: a traditional machine learning baseline model, an Indic-language transformer baseline, and a multilingual transformer-based model. We implemented TF-IDF with Linear SVM, IndicBERT, and fine-tuned XLM-RoBERTa for the classification of Tamil X (Twitter) political comments. The baseline SVM model extracts unigram and bigram TF-IDF features and combines them with a Linear Support Vector Machine for classification. However, the traditional model shows limitations in understanding contextual and semantic relationships, particularly for complex sentiment categories such as sarcasm and opinionated expressions.

In contrast, transformer-based models such as IndicBERT and XLM-RoBERTa generate contextual embeddings that improve semantic understanding of multilingual and code-mixed Tamil political text. IndicBERT achieved better performance than the baseline SVM model by effectively capturing Indic-language contextual information, while XLM-RoBERTa achieved the highest overall performance due to its stronger multilingual representation capability. To further address class imbalance, weighted cross-entropy loss was applied during transformer model training. The experimental results demonstrate that transformer-based multilingual approaches are more effective than traditional machine learning methods for low-resource political sentiment classification tasks.

References

R. Ardiansyah, H. Yuliansyah, and A. Yudhana. 2024. Twitter sentiment analysis of public space opinions using svm and tf-idf methods. *The Indonesian Journal of Computer Science*, 13(1).

Arupa Barua, Md Osama, and Ashim Dey. 2025. Cuet_novice@dravidianlangtech 2025: A bi-gru approach for multiclass political sentiment analysis of tamil twitter (x) comments. In *Proceedings of DravidianLangTech 2025*, India. DravidianLangTech.

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 on political multiclass sentiment analysis of tamil x (twitter) comments: Dravidianlangtech@naacl 2025. In *Proceedings of DravidianLangTech@NAACL 2025*.

Abhishek Gandhar, Shashi Gandhar, S. B. Kumar, Arvind Rehalia, Prakhar Priyadarshi, and Mohit Tiwari. 2024. A comparative analysis of machine learning algorithms for sentiment analysis in indian social media.

Abirami Jayaraman, Aruna Devi Shanmugam, Dharunika Sasikumar, and Bharathi B. 2025. Analysisarchitects@dravidianlangtech 2025: Machine learning approach to political multiclass sentiment analysis of tamil. In *Proceedings of DravidianLangTech 2025*, Kalavakkam, Chennai, Tamil Nadu. Sri Sivasubramaniya Nadar College of Engineering.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, and 1 others. 2020. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Haribabu Kalla. 2025. Zero-shot multilingual sentiment analysis using transformer-based models. *IJERND*, 2(02).

Naman Khurana and Vibha Prabhu. 2023. Sentiment analysis using transformer models (bert, albert, roberta, and deberta) on a smile twitter dataset. *IEEE*.

Lavanya Sambath Kumar, Asha Hegde, Bharathi Raja Chakravarthi, Hosahalli Shashirekha, Rajeswari Natarajan, Sajeetha Thavareesan, Ratnasingam Sakuntharaj, Thenmozhi Durairaj, Prasanna Kumar Kumaresan, and Charmathi Rajkumar. 2024.

Overview of second shared task on sentiment analysis in code-mixed tamil and tulu. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 62–70, St. Julian's, Malta. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, and 1 others. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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*.

Billodal Roy, Souvik Bhattacharyya, Pranav Gupta, and Niranjan Kumar M. 2025. Lexilogic@dravidianlangtech 2025: Political multiclass sentiment analysis of tamil x (twitter) comments and sentiment analysis in tamil and tulu. In *Proceedings of DravidianLangTech 2025*. Lowe's.

Mani Vegupatti, Kishore Kumar Ponnusamy, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Durairaj Thenmozhi, Prasanna Kumar Kumaresan, and Sathiyaraj Thangasamy. 2026. TamilPoliSent 2026: A Shared Task report on Multiclass Political Sentiment Analysis in Tamil. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

TriVector@DravidianLangTech 2026: Abusive Tamil Text Detection on Social Media Using Lexicon-Augmented Transformers

Oarisa Rebayet, Tahmima Hoque Eid, Fawzia Tabassum, Hasan Murad
Department of Computer Science and Engineering,
Chittagong University of Engineering and Technology, Bangladesh
{u2104129, u2104114, u2104074}@student.cuet.ac.bd, hasanmurad@cuet.ac.bd

Abstract

Abusive comment detection in low-resource languages poses significant challenges, particularly when targeting gender-based abuse on social media platforms. This work presents our system for 'Abusive Tamil text targeting women on social media' at DravidianLangTech@ACL 2026. We introduce nine handcrafted lexicon features integrated with pretrained multilingual transformer embeddings and evaluate their effectiveness in classifying Tamil online comments as abusive or non-abusive. To better understand their impact, we compare model performance with and without these lexical attributes across multiple transformer architectures. Our best-performing model, XLM-RoBERTa-Large, achieved a macro F1-score of 81.71%, securing 15th rank in the competition. The findings indicate that larger multilingual models generalize more effectively to unseen data compared to smaller domain-specific models, while the addition of lexical features yields only mild gains.

1 Introduction

Social media has become a widespread platform for global communication. At the same time, it has emerged as a primary point where people are subject to harassment. Tamil, a widely spoken Dravidian language, has seen a rise in abusive comments toward women. This may lead to severe psychological and social consequences, which emphasizes the urgent need for effective detection mechanisms. However, social media text is highly informal, characterized by frequent multilingual code-switching. This involvement of code-mixed data has introduced a complex subdomain within natural language processing (NLP). Building automated systems to detect abusive Tamil text is difficult due to the language's intricate morphological features and English code-mixing. This often causes models to misclassify neutral text as abusive.

To solve this challenge, we have taken part in Abusive Tamil Text Targeting Women on Social Media at DravidianLangTech@ACL 2026 (Rajiakodi et al., 2026), which aims to classify abusive text. This work conducts a systematic contrastive evaluation of whether statistically derived lexical features continue to add value on top of large multilingual transformers in a low-resource, code-mixed setting. Our contributions are as follows:

- We propose a lexicon-augmented transformer architecture that fuses contextual representations from pretrained language models with nine lexical features.
- We compare four pretrained models of varying scale and domain backgrounds across multiple system configurations.

2 Literature Review

The detection of abusive content in the Tamil language remains an underexplored area in NLP. Notable progress has been achieved in high-resource languages, but Tamil continues to pose significant challenges due to its complex morphological structure.

Early research on abusive language detection in Tamil relied on traditional machine learning approaches, which included models like SVM and Logistic regression (S et al., 2025). Further work compared supervised and unsupervised methods for Tamil offensive language detection (Balakrishnan et al., 2023). The DravidianLangTech shared tasks (Chakravarthi et al., 2021) played a crucial role in advancing NLP research for Tamil by providing benchmarked datasets. Transformer-based models like BERT, m-BERT, and XLNET achieved a weighted F1-score of 96% on Tamil–English code-mixed data, compared to 59% on pure Tamil text (B and Varsha, 2022).

Multiple multilingual transformer models were evaluated on the DravidianLangTech benchmark,

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 420–428
July 3, 2026 ©2026 Association for Computational Linguistics

where mBERT achieved the best performance on the monolingual Tamil dataset, obtaining a weighted F1-score of 70% (Rajalakshmi et al., 2022). It was further found that combining language-specific stemming with transformer embeddings such as XLM-RoBERTa and MuRIL meaningfully improves offensive content identification in Tamil (Rajalakshmi et al., 2023).

More recently, a variety of architectures like CNN, BiLSTM, and XLM-RoBERTa have been explored for Tamil detection (Rahman et al., 2025). A hybrid model structure has been introduced that combines transformer embeddings with attention mechanisms (Kodali et al., 2025). In the broader Dravidian language context, Ghanghor et al. (2021) presented a study on offensive language identification and meme classification in Tamil, Malayalam, and Kannada. Gong et al. (2021) handled heterogeneous abusive language by introducing a YouTube dataset with sentence-level annotations and a supervised attention model with multi-task learning. Despite these advancements, Vetagiri et al. (2024) pointed out challenges such as the lack of large annotated datasets and the informal nature of social media text, and then emphasized the need for more effective approaches to accurately detect offensive comments in Tamil.

Maheshwari et al. (2025) highlight the role of domain-aware multilingual lexicon generation in capturing context-sensitive word semantics in low-resource languages.

While most existing studies have focused on improving contextual representations using larger pretrained transformers, comparatively little attention has been paid to whether statistically derived lexical features can still offer additional benefits. By conducting such a contrastive evaluation, we explore not only the overall differences in model performance but also the extent to which lexical features continue to make meaningful contributions, especially in cases where even large contextual embeddings may miss subtle but important cues.

3 Data Description

Aiming to classify abusive comments directed toward women, the task organizers created an approximately balanced dataset comprising Tamil online comments. This dataset enables precise classification of content into the categories: Abusive and Non-Abusive. Beyond the class distribution, we also examine the linguistic characteristics of the dataset. Basic corpus statistics such as total and unique word counts, the proportion of Tamil and English tokens and the average word count provide a better understanding of the dataset.

Table 1 presents the distribution and key corpus statistics across the training, validation, and test set.

Statistic	Train	Validation	Test
Total Samples	2,921	731	913
Total Words	42,564	10,322	12,745
Unique Words	15,401	5,342	6,266
Tamil Word Count	40,192	9,729	11,939
English Word Count	1,471	394	536
Avg Word Count	14.57	14.12	13.96

Table 1: Corpus Statistics.

4 Methodology

This section presents a concise overview of the methods and approaches employed to address the problem described earlier. Following extensive analysis, the transformer-based model XLM-RoBERTa-Large exhibited the best performance for our task. Figure 1 provides a visual representation of the proposed methodology, highlighting the key steps in the approach. The implementation and source code are publicly available on GitHub.[1]

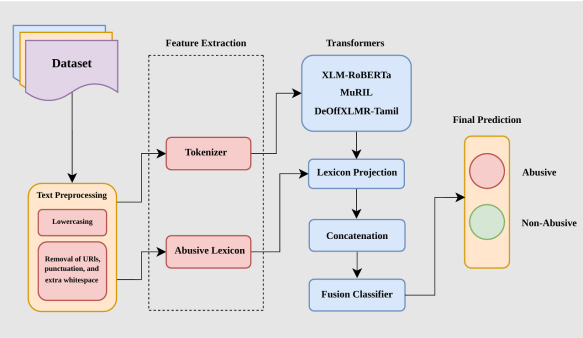


Figure 1: An abstract view of the proposed lexicon-augmented transformer system.

4.1 Text Preprocessing

Basic text preprocessing was performed to clean the dataset. This included lowercasing all text, removal of URLs, punctuation, and special characters. The same normalization procedure was applied consistently across all data splits.

[1] https://github.com/OarisaR/TriVector-abusive-tamil-detection

4.2 Lexicon Construction

We use Pointwise Mutual Information (PMI) to quantify how strongly each word is associated with the abusive class relative to the non-abusive class. For every word w occurring in at least 3 abusive documents:

$$\text{PMI}(w) = \log \frac{P(w \mid \text{Abusive})}{P(w \mid \text{Non-Abusive})} \quad (1)$$

If any word w frequently occurs in abusive comments but rarely found in non-abusive ones, the ratio exceeds 1 and PMI is positive, which indicates w is a discriminative abusive term. The top 150 words by PMI score are chosen for the abusive lexicon.

Severity Scoring

Each lexicon word is assigned a severity score that reflects *how strongly* abusive it is, by rescaling its PMI from $[-3, 3]$ to $[0, 5]$:

$$\text{sev}(w) = \text{clip}\left(\frac{\text{PMI}(w) + 3}{6} \cdot 5,\ 0,\ 5\right) \quad (2)$$

This lets the model distinguish mildly offensive words from strongly abusive ones, rather than treating all lexicon matches equally.

Lexical Feature Vector

Nine scalar features are extracted per text using the abusive lexicon and sentiment word list, as described in Table 2.

Index	Feature	Description
1	Abusive match count	Number of PMI-lexicon tokens present
2	Severity sum	Sum of severity scores of matched tokens
3	Positive word count	Count of positive-sentiment words
4	Negative word count	Count of negative-sentiment words
5	Negation count	Count of negation words
6	Intensifier count	Count of intensifier words
7	Sarcasm heuristic	1 if positive, negative, *and* negation all co-occur
8	Tamil ratio	Fraction of characters in Tamil Unicode block
9	English ratio	Fraction of characters that are ASCII letters

Table 2: Nine lexical features extracted per input

4.3 Model Architecture

We experiment with multiple transformer-based pretrained language models: XLM-RoBERTa-base and large (Conneau et al., 2019), MuRIL (Khanuja et al., 2021), and DeOffXLMR-Tamil (Saha et al., 2021). These models are specifically chosen for their strong multilingual capabilities and effectiveness in handling low-resource and Indian languages. Our primary architecture fuses transformer contextual representations with the lexicon feature vector, as illustrated in Figure 1.

Given input text x, we obtain token-level hidden states from a pretrained transformer backbone $\mathcal{T}$:

$$\mathbf{H} = \mathcal{T}(x) \in \mathbb{R}^{L \times d} \quad (3)$$

where L is the sequence length and $d = 1024$ is the hidden dimension for XLM-RoBERTa large. We apply attention-mask-weighted mean pooling:

$$\mathbf{p} = \frac{\sum_{i=1}^{L} m_i \cdot \mathbf{h}_i}{\sum_{i=1}^{L} m_i} \in \mathbb{R}^d \quad (4)$$

where $m_i \in \{0, 1\}$ denotes the attention mask. The lexicon feature vector $\mathbf{f} \in \mathbb{R}^9$, is projected through a small MLP:

$$\mathbf{v} = \text{ReLU}(\mathbf{W}_{\text{lex}}\mathbf{f} + \mathbf{b}_{\text{lex}}) \in \mathbb{R}^{64} \quad (5)$$

The pooled representation and lexicon projection are concatenated and passed through a fusion classifier:

$$\hat{y} = \mathbf{W}_2 \cdot \text{Dropout}(\text{ReLU}(\mathbf{W}_1[\mathbf{p}; \mathbf{v}] + \mathbf{b}_1)) + \mathbf{b}_2 \quad (6)$$

where $[;]$ denotes concatenation. The fusion hidden size is 128 and dropout rate is 0.2. The model is trained end-to-end with cross-entropy loss. We fine-tune all transformer backbones under a consistent training configuration, with hyperparameters summarized in Table 3.

Hyperparameters	Value
Epochs	4, 7
Learning rate	2×10^{-5}
Weight decay	0.01
Batch size	16

Table 3: Training hyperparameters.

5 Results and Discussion

Table 4 presents the test set performance across all system configurations. XLM-RoBERTa-large without lexicon augmentation achieves the highest macro F1 score of 81.71%, followed by XLM-RoBERTa-base with lexicon augmentation at 80.81%. While the large model offers slightly better performance, it incurs a significantly higher computational cost due to its increased size.

In resource-constrained settings, XLM-RoBERTa-base with lexical augmentation provides a balanced trade off, achieving 81.03% macro F1, only 0.68% lower than the large model, while remaining substantially more efficient.

System	Epochs	Approach	Macro P	Macro R	Macro F1
MuRIL-base	4	−Lex	79.51	79.28	79.32
		+Lex	78.50	78.30	78.34
		Δ	-1.01	-0.98	-0.98
	7	−Lex	78.74	78.70	78.72
		+Lex	79.07	79.10	79.07
		Δ	+0.33	+0.40	+0.35
DeOffXLMR-Tamil	4	−Lex	78.31	78.25	78.27
		+Lex	78.95	78.98	78.96
		Δ	+0.64	+0.73	+0.69
	7	−Lex	77.51	75.72	75.60
		+Lex	77.56	77.46	77.49
		Δ	+0.05	+1.74	+1.89
XLM-R-base	4	−Lex	79.76	78.04	78.00
		+Lex	80.81	80.81	80.81
		Δ	+1.05	+2.77	+2.81
	7	−Lex	81.00	79.50	79.51
		+Lex	79.51	79.47	79.48
		Δ	-1.49	-0.03	-0.03
XLM-R-large	4	−Lex	81.32	80.61	80.66
		+Lex	79.51	79.54	79.51
		Δ	-1.81	-1.07	-1.15
	7	−Lex	81.84	81.82	**81.71**
		+Lex	78.84	78.85	78.84
		Δ	-3.00	-2.97	-2.87

Table 4: Performance of various methods, reported in %

XLM-R-large Achieves Best Performance. XLM-R-large (−Lex) which is trained for 7 epochs achieves the highest Macro F1 of **81.71**% across all systems and settings. This result highlights the strength of large-scale model for abusive Tamil detection, even without additional lexical augmentation.

Lexical Augmentation Shows Inconsistent Gains. The impact of lexical augmentation (+ Lex) varies across models and training durations. For XLM-R-base at 4 epochs, lexical features yield a gain of +2.81% Macro F1, the largest positive increase observed in the entire table. In contrast, both XLM-R-large configurations suffer from lexical augmentation, with drops of 1.15% and 2.87% at 4 and 7 epochs respectively, indicating that larger models already encode sufficient linguistic knowledge and lexical features introduce noise rather than signal.

Models Exhibit Varying Epoch Sensitivity. MuRIL-base remains relatively stable across both augmentation settings. DeOffXLMR-Tamil exhibits the most notable epoch-sensitive behavior: at 7 epochs without lexical features its Macro F1 drops to 75.60%, but lexical features recovers it substantially to 77.49%. XLM-R-base peaks at 7 epochs without augmentation (79.51%), while adding lexical features at that epoch only marginally affects performance (0.03%).

Table 5 presents a few examples from our best-performing system.

Text	Actual	Predicted
அப்பா,அம்மா, அந்த இன்டர்வியூ பண்றவ வக்கிரம்புடிச்சவ.. எல்லோருமே செருப்படி வாங்க தகுதியானவங்க.. அதிலயும் அந்த பொண்ணு.. யப்பா சாமி... முடியலடா. (Appa, amma, that interview-giving crooked fellow... Everyone come slap him, those who deserve it... And that woman... ya pa sami... can't take it da.)	Abusive	Abusive
நேர் கொண்ட பார்வை அடுத்தவங்க மானத்தை வாங்கி தெருவில் விடுவதே உங்க வேலையா போச்சி (So this "straight-forward look" of yours now means grabbing someone else's honour and leaving it in the street? Has that become your job or what?)	Non-Abusive	Non-Abusive
இவ என்ன சட்டம் படிச்ச நீதிபதியா???எல்லாருக்கு நீதி சொல்ல முண்ட,இவளுக்கு ஒருநாள் நிகழ்ச்சி நடத்தப்படும் (Who is she, some law-studied judge??? To sit and tell justice for everyone... For her too one day a program will be conducted.)	Abusive	Abusive
ஒடி பேயிரு கையில் மாடிராத நான் இலங்கை இருக்கன் டிக்கெட் போட்டு வாந்து ஓதய்பன் யார் நீ விஜய் பாத்தி பேசா (Run away. If I get my hands on you, I won't spare you. I'm in Sri Lanka. I'll buy a ticket and come there. Who do you think you are talking about Vijay like that?)	Non-Abusive	Abusive
ஜி.பி .முத்து mind voice. செத்த பயலே நார பயலே.. செத்த மூதேவி ஆளும் மூஞ்சியும் பாரு.. (G.P. Muthu (mind voice): You useless rascal, you rotten fellow… that wretched woman just look at her, look at that face…)	Abusive	Non-Abusive

Table 5: Examples of XLM-RoBERTa-large model predictions.

6 Conclusion

This work compares several transformer-based models for detecting abusive comments in Tamil. We evaluated different system settings across four pretrained architectures. Among them, XLM-RoBERTa-large without any lexicon enhancement achieved the highest macro F1-score of 81.71%, while XLM-RoBERTa-base combined with lexical features reached 80.81%. The findings indicate that larger models tend to generalize better to unseen examples, while handcrafted lexical features are useful for mid-sized models. However, PMI-based lexicon construction carries a risk of named-entity bias, which can hurt precision in certain cases.

Limitations

While the system performs well, it still has practical limitations. Tamil social media text is highly informal, making cultural context difficult to capture automatically. The PMI-based lexicon construction may cause frequently targeted public figures to acquire high PMI scores through corpus co-occurrence rather than inherent abusiveness. Filtering named entities prior to severity scoring is a direction for mitigating this. Additionally, the lexical features are surface-level and largely language-agnostic; incorporating Tamil-specific phenomena such as morphological decomposition and honorific misuse patterns could yield richer signals. Future work should also explore finer-grained labels and stronger generative architectures.

Ethical Considerations

This study utilizes publicly available Tamil social media data released through the shared task, adhering to ethical research standards. User identities were neither tracked nor disclosed, and the data were used exclusively for academic purposes. While our system aims to support safer online environments, automated detection models carry inherent risks, including bias and misclassification, and should not be deployed without human oversight. Particularly, lexicon construction methods based on corpus co-occurrence statistics may encode dataset-level biases, potentially disadvantaging the very demographic groups the system is designed to protect.

References

Bharathi B and Josephine Varsha. 2022. SSNCSE NLP@TamilNLP-ACL2022: Transformer based approach for detection of abusive comment for Tamil language. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 158–164, Dublin, Ireland. Association for Computational Linguistics.

Vimala Balakrishnan, Vithyatheri Govindan, and Kumanan N. Govaichelvan. 2023. Tamil offensive language detection: Supervised versus unsupervised learning approaches. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(4).

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Navya Jose, Anand Kumar M, Thomas Mandl, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Hariharan R L, John P. McCrae, and Elizabeth Sherly. 2021. Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 133–145, Kyiv. 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. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Nikhil Ghanghor, Parameswari Krishnamurthy, Sajeetha Thavareesan, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. 2021. IIITK@DravidianLangTech-EACL2021: Offensive language identification and meme classification in Tamil, Malayalam and Kannada. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 222–229, Kyiv. Association for Computational Linguistics.

Hongyu Gong, Alberto Valido, Katherine M. Ingram, Giulia Fanti, Suma Bhat, and Dorothy L. Espelage. 2021. Abusive language detection in heterogeneous contexts: Dataset collection and the role of supervised attention. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17):14804–14812.

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, Vishnu K Subramanian, and Partha Pratim Talukdar. 2021. Muril: Multilingual representations for indian languages. *ArXiv*, abs/2103.10730.

Rohith Gowtham Kodali, Durga Prasad Manukonda, and Maharajan Pannakkaran. 2025. byteSizedLLM@DravidianLangTech 2025: Abusive Tamil and Malayalam text targeting women on social media using XLM-RoBERTa and attention-BiLSTM. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 80–85, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Ayush Maheshwari, Atul Kumar Singh, N J Karthika, Krishnakant Bhatt, Preethi Jyothi, and Ganesh Ramakrishnan. 2025. LexGen: Domain-aware multilingual lexicon generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7364–7375, Vienna, Austria. Association for Computational Linguistics.

MD.Mahadi Rahman, Mohammad Minhaj Uddin, and Mohammad Shamsul Arefin. 2025. CUET_Ignite@DravidianLangTech 2025: Detection of abusive comments in Tamil text using transformer models. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages

392–397, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Ratnavel Rajalakshmi, Ankita Duraphe, and Antonette Shibani. 2022. DLRG@DravidianLangTech-ACL2022: Abusive comment detection in Tamil using multilingual transformer models. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 207–213, Dublin, Ireland. Association for Computational Linguistics.

Ratnavel Rajalakshmi, Srivarshan Selvaraj, Faerie Mattins R., Pavitra Vasudevan, and Anand Kumar M. 2023. HOTTEST: Hate and offensive content identification in Tamil using transformers and enhanced STemming. *Computer Speech & Language*, 78:101464.

Saranya Rajiakodi, Shunmuga Priya Muthusamy Chinnan, Rajalakshmi R., Kathiravan Pannerselvam, Bhuvaneswari Sivagnanam, Jananayagan V, Charmathi Rajkumar, R Ramesh Kannan, and Bharathi Raja Chakravarthi. 2026. From Comments to Harm: A Findings Report on Abusive Tamil Text Targeting Women on Social Media Shared Task- DravidianLangTech@ACL 2026. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Varun Balaji S, Bojja Revanth Reddy, Vyshnavi Reddy Battula, Suraj Nagunuri, and Balasubramanian Palani. 2025. CoreFour_IIITK@DravidianLangTech 2025: Abusive content detection against women using machine learning and deep learning models. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 655–660, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Debjoy Saha, Naman Paharia, Debajit Chakraborty, Punyajoy Saha, and Animesh Mukherjee. 2021. Hate-alert@DravidianLangTech-EACL2021: Ensembling strategies for transformer-based offensive language detection. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 270–276, Kyiv. Association for Computational Linguistics.

Advaitha Vetagiri, Gyandeep Kalita, Eisha Halder, Chetna Taparia, Partha Pakray, and Riyanka Manna. 2024. Breaking the silence detecting and mitigating gendered abuse in hindi, tamil, and indian english online spaces. *ArXiv*, abs/2404.02013.

A Appendix

A.1 Error Analysis

We conducted an error analysis on the misclassified outputs of XLM-RoBERTa-Large, examining 167 misclassified samples, where there were 101 false positives (FP) and 66 false negatives (FN).

False positives occur when the model relies on surface-level cues such as negative sentiment or personal aggression, rather than pragmatic intent. Table 6 illustrates this tendency with representative examples.

Tamil Text	English Translation	Failure Pattern
ஒடி பேயிரு கையில் மாடிராத நான் இலங்கை இருக்கன் டிக்கெட் போட்டு வாந்து ஓதய்பன் யார் நீ விஜய் பாத்தி பேசா	"Run away. If I get my hands on you, I won't spare you. I'm in Sri Lanka. I'll buy a ticket and come there. Who do you think you are talking about Vijay like that?"	Aggressive fan defense misinterpreted as abuse
எல்லாம் பணம் பண்ணும் வேலை நாம ஒரு நாள் கூலி வேலை செய்து பொலைக்கிறோம் இது போன்ற ஜென்மம் கேவலமா இருந்து சம்பாதிக்குது	"Everything is about money. We struggle surviving on daily wages, while people like this earn money living shamefully"	Social critique misinterpreted as abuse

Table 6: Representative false positive examples with identified failure pattern.

False negatives arise when the model fails to recognize abuse that is conveyed through cultural expressions or cross-script markers. In such cases, the absence of explicit offensive language leads the model to misclassify abusive content as non-abusive, reflecting a broader lack of contextual grounding, as shown in Table 7.

Tamil Text	English Translation	Failure Pattern
உங்கல பார்தா சிரிப்புதாண் வருது லஷ்மி மேடம்	"Just seeing you makes me laugh, Madam Lakshmi"	Mockery disguised as polite addressing
Suganthi உனக்கு வீட்ல குழந்தையை வச்சிருக்க கார்த்திக் உன்ன கட்டி புடிச்சி விளையாடறான் கொஞ்சம் கூட வெக்கம் இல்ல thuuuu	"Suganthi, you have a child at home yet Karthik is hugging and playing with you, not even an ounce of shame, thuuuu"	Moral shaming with gendered insult

Table 7: Representative false negative examples with identified failure pattern.

Figure 2 presents the confusion matrix for XLM-RoBERTa-Large, illustrating the overall distribution of classification outcomes. The higher frequency of false positives compared to false negatives indicates that the model is sensitive to surface-level aggression, flagging contextually grounded expressions as abusive while struggling to detect abuse that is culturally embedded.

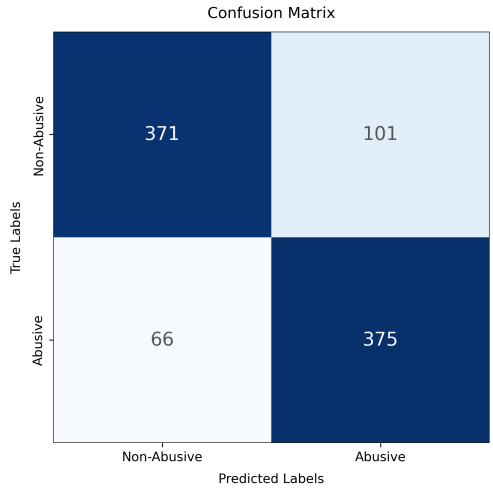


Figure 2: Confusion matrix for XLM-RoBERTa-Large.

A.2 Lexical Feature Analysis

A.2.1 PMI Lexicon Inspection

To assess whether the PMI-derived lexicon captures genuine abusive indicators or dataset-specific biases, Table 8 presents the top 15 words ranked by PMI score along with their assigned severity values.

Rank	Word	PMI	Severity
1	குஷ்பு	3.15	5.00
2	சின்மயி	3.06	5.00
3	தேவடியா	2.77	4.81
4	குஷ்பூ	2.77	4.81
5	சுந்தர்	2.70	4.75
6	பால்	2.55	4.62
7	டாக்டர்	2.46	4.55
8	அவுத்து	2.36	4.47
9	கிழவி	2.36	4.47
10	கனிமொழி	2.26	4.38
11	மக்களுக்கு	2.26	4.38
12	நாதாரி	2.26	4.38
13	doctor	2.26	4.38
14	பட்ட	2.14	4.28
15	பெண்களுக்கு	2.14	4.28

Table 8: Top 15 words by PMI score with severity values.

The lexicon reveals two distinct word categories. The first comprises genuinely abusive Tamil terms such as தேவடியா (prostitute), அவுத்து (danger/risk), நாதாரி (worthless person), and கிழவி (old woman), which are well-established slurs targeting women. The second category raises concern: named individuals (குஷ்பு (Khushboo), சின்மயி (Chinmayi), கனிமொழி (Kanimozhi)) rank highest by PMI, reflecting targeted harassment campaigns against specific public figures in the dataset rather than inherently abusive vocabulary. Furthermore, context-dependent words such as டாக்டர் (doctor) and மக்களுக்கு (for the people) acquired high PMI scores due to their frequent co-occurrence with abusive comments in the corpus, indicating dataset-specific bias rather than genuine abusiveness.

A.2.2 PMI Threshold Sensitivity

To evaluate the robustness of our lexicon construction, we examined how the PMI score distribution changes across different top-N cutoffs, as shown in Table 9.

Top-N	Min PMI	Max PMI	Mean PMI
50	1.8540	3.1533	2.1482
100	1.6717	3.1533	1.9633
150	1.5663	3.1533	1.8645
200	1.4485	3.1533	1.7642

Table 9: PMI score distribution across different lexicon size cutoffs.

The maximum PMI remains stable at 3.15 across all thresholds, while the mean decreases gradually from 2.15 at top-50 to 1.76 at top-200. This gradual degradation confirms that our chosen cutoff of 150 words represents a reasonable trade-off, retaining highly discriminative terms while limiting the inclusion of low-confidence associations.

A.2.3 Lexical Feature Importance

To understand which of the nine lexical features contributed most to the model's decisions, we analyzed the mean absolute weights of the learned lexicon projection layer, as reported in Table 10.

Rank	Feature	Importance
1	Positive word count	0.1817
2	Negation count	0.1789
3	English ratio	0.1784
4	Severity sum	0.1769
5	Tamil ratio	0.1739
6	Abusive match count	0.1704
7	Intensifier count	0.1668
8	Negative word count	0.1652
9	Sarcasm heuristic	0.1386

Table 10: Lexical feature importance derived from learned projection weights.

The importance scores are relatively uniform across features, indicating that the projection layer learns a distributed representation rather than relying on any single signal. Positive word count (0.1817) and Negation count (0.1789) rank highest, suggesting that sentiment polarity reversal, where positive words are negated is a useful surface cue

for abuse detection. English ratio (0.1784) ranks third, reflecting the code-mixed nature of Tamil social media text. Notably, Sarcasm heuristic received the lowest weight (0.1386), suggesting the simple co-occurrence heuristic was insufficient to capture nuanced sarcastic abuse.

A.2.4 Instance-Level Analysis of Lexical Feature Impact

Since XLM-RoBERTa-base exhibited the highest lexical gain, its predictions under two settings were compared to analyze the impact of lexical features. The comparison was performed across all 913 test instances, of which 89 predictions (9.75%) differed between the two settings, as summarized in Table 11.

True Label	−Lex Pred	+Lex Pred	Count
Abusive	Non-Abusive	Abusive	16
Non-Abusive	Abusive	Non-Abusive	40
Abusive	Abusive	Non-Abusive	21
Non-Abusive	Non-Abusive	Abusive	12
Net benefit (corrections − errors)			**+23**

Table 11: Breakdown of the test predictions that changed between −Lex and +Lex setting.

LEX recovers missed abuse (16 cases). These instances contained Tamil specific abusive vocabulary that XLM R base multilingual embeddings did not sufficiently prioritize when considered in isolation. The comment:

> செத்த மூதேவி ஆளும் மூஞ்சியும் பாரு ("Look at that wretched woman's face and demeanor")

was correctly reclassified as abusive once மூதேவி ("wretched woman," a strong Tamil gendered slur) triggered a high severity score in the lexicon. Similarly, the comment:

> டிக் டாக் லா இந்த அரிப்பு தாங்காம பொண்ணுங்க தான் அவுத்து போட்டு ஆடி ("These women on TikTok, unable to contain themselves, stripping and dancing")

contains அவுத்து ("stripping," a sexually objectifying term, ranked 8th in the PMI table with a severity score of 4.47). Without LEX, the transformer model interpreted this as social commentary, whereas the lexicon based severity signal correctly captured the underlying abusive intent.

LEX fixes false alarms (40 cases). This is the most significant effect of lexical augmentation. Without LEX, 40 neutral comments were misclassified as abusive due to their aggressive surface tone. LEX corrected these cases by showing low abusive match counts and high Tamil ratio scores, indicating genuine social discourse rather than targeted abuse. The comment:

> சுகந்தி பார்க்க நல்ல விதமாக அறிவான பெண்ணாக இருக்கிறார் ("Suganthi appears to be a sensible and intelligent woman")

is a straightforward compliment. The base model incorrectly flagged it as abusive because the named entity சுகந்தி ("Suganthi") frequently co-occurs with abusive comments in the training data. LEX reversed this prediction through its high positive word count signal and zero abusive match count. Similarly, the comment:

> மனுஷங்களாடா நீங்க..... ரொம்ப தரம் தாழ்ந்து விமர்சிக்கிரீங்க..... மூன்று மாத குழந்தையா இருந்த போதே தந்தைய இழந்துட்டு... (Are you even human..... you are criticizing so cheaply..... She lost her father when she was just a three-month-old baby)

is a defense of a woman being harshly criticized rather than an attack. The aggressive tone led the transformer model to misinterpret the context, whereas LEX identified no abusive lexical matches and detected a positive sentiment signal, correctly classifying the instance as Non Abusive.

LEX introduces false negatives (21 cases). These cases reveal a fundamental limitation of surface level lexical features, where culturally embedded abuse is expressed through gendered sarcasm without triggering any PMI lexicon signal. The comment:

> லோக்கல் ஐட்டம்மும் இண்டர்நேஷல் ஐட்டம்மும் உரையாடிய போது.... ("When the local item and the international item were talking....")

uses ஐட்டம் ("item," an objectifying term in Tamil cultural discourse) to demean women. Since this word carries no explicit PMI-lexicon match, LEX assigned a near-zero abuse score. Likewise:

> ரெண்டு பேரும் பத்தினி மாதிரி பேசுராலுக (“Both of them speak as if they are chaste women...”)

employs பத்தினி (“chaste woman”) sarcastically as a tool of moral shaming. The sarcasm heuristic failed to activate because the comment does not simultaneously contain the required positive, negative, and negation markers. Consequently, LEX suppressed the transformer’s accurate abuse identification.

LEX introduces false positives (12 cases). Named entity bias in the PMI lexicon is the primary driver of these errors. Public figures who are frequent targets of online harassment, such as குஷ்பு and சின்மயி, acquired high PMI scores through repeated co-occurrence with abusive comments in the training corpus. Their mere mention in neutral or supportive contexts consequently triggered abuse classification. The most striking example is:

> பெண் சுதந்திரம், பெண் கல்வி, பெண் உரிமை (“Women’s freedom, women’s education, women’s rights”)

a feminist slogan that was incorrectly classified as abusive. The word பெண் (“woman”) itself acquired a spurious abusive association from corpus statistics, representing the most direct evidence that PMI-based lexicons risk encoding dataset-level bias against the very demographic they are designed to protect.

Similarly, the comment:

> இந்த பொண்ணுக்கு புத்தி சொல்லி *encourage, motivation* குடுத்து சாதித்து தன் சுயமரியாதையுடன் நிற்க பயிற்சி கொடுங்க.... *she is innocent but, mind got contaminated.....* (“Give advice to this girl, encourage and motivate her, and train her to achieve success and stand with self-respect... she is innocent, but her mind has become contaminated”)

shows genuine supportive advice for a young woman. But the presence of mild criticism triggered PMI lexicon matches, causing incorrect abusive prediction.

Overall, LEX improved the model’s predictions, leading to an increase in Macro F1 score from the no-lexicon baseline to the enhanced system. These results indicate that lexical augmentation is particularly effective in mitigating surface-level aggression, especially in mid-sized transformer models that may struggle to distinguish between aggressive yet neutral expressions and genuinely targeted abuse. However, two systematic failure modes persist. The first involves culturally embedded abusive content expressed through gendered metaphor and moral shaming, which lacks explicit lexical markers. The second concerns named entity bias, where frequently targeted individuals and demographic terms such as பெண் (“woman”) develop unintended associations with abusive content due to corpus-level co-occurrence patterns.

TriVector@DravidianLangTech 2026: Depression Detection from Tamil and Malayalam Speech with Speaker-Independent Evaluation using MFCC and Wav2Vec2

Tahmima Hoque Eid, Fawzia Tabassum, Oarisa Rebayet, Hasan Murad
Department of Computer Science and Engineering,
Chittagong University of Engineering and Technology, Bangladesh
{u2104114, u2104074, u2104129}@student.cuet.ac.bd, hasanmurad@cuet.ac.bd

Abstract

Depression is a major mental health concern that can be reflected through subtle changes in speech patterns, prosody, and vocal characteristics. In low-resource and multilingual settings, depression detection from speech may become particularly more challenging. In this work, we present our system for the Shared Task on Depression Detection from Malayalam and Tamil. We explored both handcrafted acoustic features (MFCC) and pretrained speech representations (Wav2Vec2) for depression detection, along with a simple fusion strategy to examine their complementary strengths. Our observations showed that Wav2Vec2 generalized better for Malayalam, whereas for Tamil, a validation-tuned probability fusion performed best. The final system achieved macro-F1 scores of 99.5% for Malayalam and 88.6% for Tamil, securing 3rd place in both tasks.

1 Introduction

Depression is a common mental health condition characterized by persistent low mood and loss of interest in daily activities. It is also accompanied by fatigue, sleep disturbances, and cognitive difficulties (James et al., 2018). Despite its widespread impact, depression remains overlooked and undertreated even today. A cross-sectional study in India reported that although nearly two-thirds of adults with depression accessed health services, about 97% remained undiagnosed (Teufel et al., 2024). Speech can capture subtle cognitive and emotional changes. Hence, it is a promising non-invasive marker for detecting depressive states. However, the majority of this field's research focuses on high-resource languages, leaving many regional languages unexplored. Although Malayalam and Tamil are widely spoken, they have limited speech-based mental health resources. Multimodal systems can also be effective in this regard, but speech-based approaches are simpler for telehealth and low-resource contexts (Rezaee, 2026).

This study is conducted as part of the Depression Detection in Dravidian Languages (DD-DL) Shared Task (Lal et al., 2026). The organizers have provided a curated set of speech recordings collected under controlled conditions. The task comprises two subtasks[1]:

(i) Depression detection in Tamil.

(ii) Depression detection in Malayalam.

Similar dataset and task formulations are described in the DraviMood study, which explores speech-based depression classification in Dravidian languages using feature fusion and deep learning techniques (Kritika et al., 2025).

To solve this challenge, we developed a depression detection system that integrates handcrafted acoustic features with self-supervised speech representations. We used MFCC features to capture low-level acoustic cues, which provide compact spectral representations of speech and are computationally efficient. We also fine-tuned Wav2Vec 2.0 to learn higher-level patterns from raw audio. Finally, we explored a combined approach by fusing the predictions of both models at the probability level to leverage their complementary strengths. Our pipeline achieved macro-F1 scores of 99.5% for Malayalam and 94.4% for Tamil.

The main contributions of this work are as follows:

- We created speaker-aware, leakage-free splits using stratified grouping for more generalization to unseen speakers.
- A depression detection model was developed using MFCC features and fine-tuned Wav2Vec 2.0.
- We applied validation-tuned probability fusion and threshold calibration to enhance reliability and class balance.

[1] https://www.codabench.org/competitions/11331/

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 429–435
July 3, 2026 ©2026 Association for Computational Linguistics

Detailed implementation information is available in our GitHub repository.[2]

2 Related Work

Early studies on speech-based depression detection relied on manually extracted speech features combined with traditional machine learning classifiers. Janardhan and Kumaresh (2022) utilized eGeMAPS features with Fisher score-based feature selection and dynamic ensemble strategies, which improved classification performance.

With the rise of deep learning, researchers shifted towards automatic representation learning from raw and spectral speech signals. Kim et al. (2023) demonstrated improved performance over conventional models by developing a smartphone-based CNN framework using log-Mel spectrograms. Gupta et al. (2024) proposed a deep convolutional attention cascaded LSTM framework for speech-based depression detection, combining feature extraction, optimization-based feature selection, and deep classification. To better capture hierarchical and long-range contextual information,Yin et al. (2023) further explored transformer-driven and parallel CNN designs.

More recent studies have focused on multilingual and low-resource scenarios. Mathew et al. (2024) introduced a bilingual spontaneous speech corpus to analyze emotional valence–arousal patterns across English and Malayalam, while related read-speech analyses were further examined by Daly and Olukoya (2025). Binu et al. (2024) investigated language-agnostic acoustic markers, suggesting that certain vocal biomarkers may generalize across linguistic settings. In low-resource environments, Zhang et al. (2024) utilized transfer learning with pretrained wav2vec 2.0, reporting notable improvements in F1 scores through contextual speech representations. Prior work in Dravidian language analysis has shown that transformer-based models such as multilingual BERT, RoBERTa, and MuRIL effectively capture linguistic patterns for sentiment and abusive content detection (Premjith et al., 2024)

Although advanced deep learning and transfer learning techniques have shown strong performance, most studies often overlook issues such as speaker leakage and data imbalance. In low-resource settings, resource scarcity remains a major hindrance to building stable and generalizable models (Anilkumar et al., 2026). Moreover, prior works rarely explore hybrid approaches combining acoustic and self-supervised features, highlighting the need for more adaptable frameworks.

3 Data Description

We used the dataset released for the Shared Task on Depression Detection from Malayalam and Tamil Speech Data – DravidianLangTech@ACL 2026 (Lal et al., 2026). The recordings were collected under controlled acoustic conditions to ensure consistent quality. Each utterance has an average duration of 2–5 seconds. All dataset statistics are summarized in Table 1.

Dataset Characteristics	Malayalam	Tamil
Training Samples	1688	1374
Test Samples	200	160
Depressed Samples	888	534
Depressed Speakers	3	4
Non-depressed Speakers	5	5
Recording Rate (Depressed)	16 kHz	16 kHz
Recording Rate (Non-depressed)	48 kHz	48 kHz

Table 1: Comprehensive dataset statistics for Malayalam and Tamil depression detection tasks.

4 Methodology

4.1 Audio Preprocessing

All recordings were standardized to 16 kHz mono for consistency. Silence trimming (top_db = 30) was applied in the Tamil pipeline, while near-silent Malayalam recordings were automatically detected and removed. To stabilize feature extraction, we applied amplitude normalization and clipping. We also discarded fully trimmed or empty Tamil recordings.

4.2 Speaker-Aware Leak-Safe Data Splitting

To prevent speaker leakage and reduce overfitting in limited-data settings, speaker identities were parsed from filenames and used for speaker-aware splitting. For Malayalam, StratifiedGroupKFold (5 folds) with speaker/group constraints was used, selecting the most class-balanced fold as validation. For Tamil, a speaker-stratified split ensured no speaker overlap between train and validation sets. Final splits were saved as leak-safe CSV files.

4.3 Overview of Experimental Models

4.3.1 MFCC-Based Acoustic Modeling

We used MFCC features to capture spectral patterns associated with depression-related speech

[2] https://github.com/tahmima114/DD-MT

changes. First, the audio was converted to 16 kHz and then transformed into fixed-size MFCC representations. Malayalam used 20 MFCCs (20×120 from 6-s audio) with train-only time–frequency masking and a 2-layer MLP classifier (serving as a computationally efficient baseline), whereas Tamil used 40 MFCCs (40×320) with silence trimming (top_db = 30) and a lightweight CNN to better capture time–frequency patterns. Finally, class imbalance was handled using weighted loss (and `WeightedRandomSampler` for Tamil). Training used AdamW optimization with early stopping based on validation macro-F1.

4.3.2 Wav2Vec2-Based Representation Learning

Wav2Vec2 was implemented for our model to learn high-level speech patterns directly from raw audio. For Malayalam, a Wav2Vec2-Base model processed 6-second segments with light training-time augmentation (speed, pitch, and noise). The classification layers were trained using class-weighted loss and gradient accumulation based on validation macro-F1. For Tamil, the Wav2Vec2-XLS-R-300M model was fine-tuned in two stages. In stage one, only the classifier was trained, whereas in stage two, the last two transformer layers were unfrozen to enable partial fine-tuning. Comparable hyperparameters are summarized in Table 2.

Hyperparameter	Malayalam	Tamil
Initial Encoder Setting	Frozen	Frozen
Training Strategy	Single-stage	Two-stage fine-tuning
Epochs (Stage 1)	15	8
Epochs (Stage 2)	–	8
Learning Rate (Stage 1)	3×10^{-4}	5×10^{-5}
Learning Rate (Stage 2)	–	1×10^{-5}
Mixed Precision (fp16)	Disabled	Enabled (Stage 1)
Early Stopping Patience	3	2

Table 2: Comparable Wav2Vec2 hyperparameters for Malayalam and Tamil.

4.3.3 Ensemble Decision Fusion

MFCC captures low-level acoustic features like spectral envelope and energy, while Wav2Vec2 learns higher-level phonetic representations from raw audio. Their combination integrates acoustic and phonetic representations for improved modeling. Accordingly, MFCC and Wav2Vec2 predictions were fused at the probability level using a weighted average of softmax outputs, with weights tuned exclusively on the validation set to avoid data leakage. The overall pipeline is illustrated in Fig. 1. For Malayalam, a small set of candidate weights was evaluated using validation macro-F1 to select the optimal fusion weight. For Tamil, a finer grid search was performed to jointly optimize the fusion weight and decision threshold using an F2-score objective, followed by a small threshold adjustment (+0.03) to improve the precision–recall balance.

5 Results and Analysis

This section presents the results of the depression detection task for Malayalam and Tamil speech. We evaluated model performance using precision, recall, and F1-score, emphasizing macro-averaged F1 (Macro-F1) to ensure balanced assessment across classes. The Macro-F1 score is computed as the average of the F1-scores of all classes:

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

Here C denotes the total number of classes, and P_i and R_i represent the precision and recall of class i, respectively.

5.1 Task 1: Depression detection in Tamil

For the Tamil task, the MFCC–CNN model reached a macro-F1 of 77.8%, showing moderate ability to capture depression-related spectral cues. In contrast, the standalone Wav2Vec2 model performed much lower (macro-F1 = 48.1%) due to class imbalance effects and biased predictions. Fusion weights were optimized on the validation set using an F2 objective with threshold calibration. Since F2 emphasizes recall, the optimization favored probability distributions that preserved higher recall. On the validation set, incorporating MFCC probabilities did not increase recall beyond that achieved by Wav2Vec2 alone. Consequently, the final system relied on Wav2Vec2 posterior probabilities. But to improve class balance, we applied decision-threshold tuning to the depression-class, leading to a final macro-F1 of 94.4% as shown in Table 3.

Model	Prec. (%)	Rec. (%)	F1 (%)
MFCC–CNN	85.09	78.75	77.75
Wav2Vec2	77.03	57.50	48.13
Validation-tuned Ensemble	94.44	94.38	94.37

Table 3: Performance comparison of Tamil depression detection models on the test set.

5.2 Task 2: Depression detection in Malayalam

Initial experiments showed strong performance from both models, with the MFCC–MLP achiev-

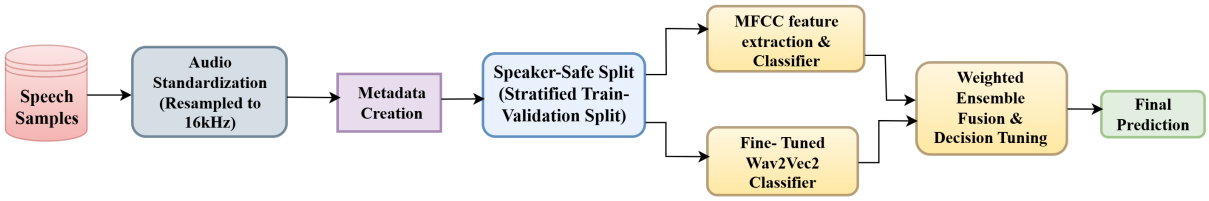
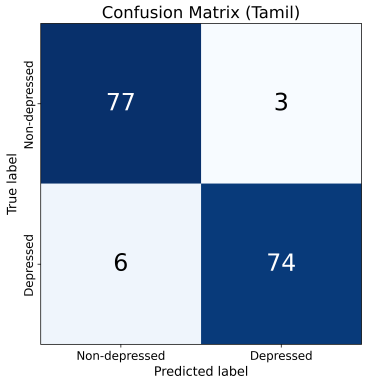


Figure 1: Overview of the proposed depression detection pipeline

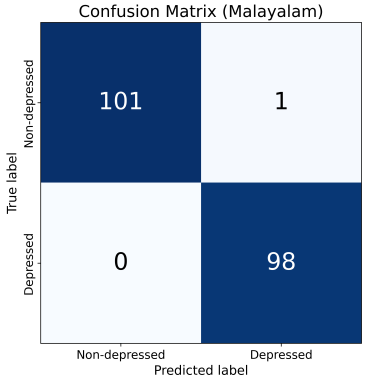
ing a macro-F1 of 99% and the Wav2Vec2 model obtaining 98.5% on the Malayalam test set (Table 4). Similar to the Tamil subtask, weight tuning for Malayalam also indicated that Wav2Vec2 representations generalized more reliably than MFCC-based predictions. Since fusion was implemented via a convex combination of posterior probabilities followed by argmax selection, introducing a non-zero MFCC weight modifies the resulting probability distribution and can shift class decisions. Validation results showed that increasing the MFCC contribution did not improve macro-F1 and slightly degraded performance. So, we retrained the Wav2Vec2 model on the combined training and validation data using light augmentation and class-weighted optimization. This slight refinement improved the macro-F1 score to 99.5% on the test set.

Model	Precision (%)	Recall (%)	Macro-F1 (%)
MFCC–MLP	99.04	98.98	99.00
Wav2Vec2	98.51	98.49	98.50
Refined Wav2Vec2	99.49	99.51	99.50

Table 4: Performance comparison of Malayalam depression detection models on the test set.



(a) Tamil



(b) Malayalam

Figure 2: Confusion matrices of the final models on the test sets.

5.3 Error Analysis

To further analyze model behavior, the confusion matrices of the final systems are shown in Fig. 2. The Malayalam model achieved almost perfect class separation with only one false positive. But the Tamil model showed slightly higher confusion between classes with 3 false positives and 6 false negatives.

Fusion weights indicated minimal complementarity between MFCC and Wav2Vec2 predictions in both languages. Malayalam benefited primarily from representation refinement, whereas Tamil required decision-threshold calibration to mitigate class bias and achieve balanced performance.

6 Conclusion

This work proposed a speech-based depression detection framework for Tamil and Malayalam by combining MFCC-based neural models and pre-trained Wav2Vec2 representations. MFCC features provided reliable spectral cues, while Wav2Vec2 captured higher-level speech patterns from raw audio. We also explored ensemble fusion to leverage complementary strengths but validation results indicated that Wav2Vec2 generalized better to unseen data, which led us to a refined training strategy. The final systems achieved strong performance on the test sets, reaching a macro-F1 of 99.5% for Malayalam and 94.4% for Tamil. Throughout the task, we tried to address major obstacles such as limited data, class imbalance, and speaker variability by implementing speaker-aware splitting, augmentation, and class-weighted optimization.

Limitations

Despite strong results, the system has several limitations. The datasets are relatively small and collected under controlled conditions with limited variability, which may restrict generalization to diverse speakers, environments, and demographic groups. Although speaker-aware splitting reduces leakage, some residual biases may still remain. A two-layer MLP is used as a lightweight baseline rather than a state-of-the-art model to evaluate MFCC effectiveness under low-resource settings. Similarly, most Wav2Vec2 encoder layers are frozen to ensure training stability, but this may limit task-specific adaptation. While recent foundation models such as HuBERT and Whisper offer stronger generalization through large-scale pretraining, their high computational and data requirements make them unsuitable for this setting. From a clinical perspective, the system is intended for preliminary screening or decision support rather than diagnosis, and therefore requires further validation on more diverse populations before real-world deployment.

Ethical Considerations

This work was conducted following the highest ethical research practices. The study aims to support research on depression detection while respecting privacy and cultural diversity. We acknowledge the sensitive nature of mental health analysis and have taken steps to reduce bias and ensure fair model behavior. The findings are intended to assist research and awareness efforts, not to replace professional clinical judgment.

References

A. Anilkumar, G. Jyothish Lal, B. Premjith, and B. R. Chakravarthi. 2026. DravLangGuard: A Multimodal Approach for Hate Speech Detection in Dravidian Social Media. In *Speech and Language Technologies for Low-Resource Languages*, volume 2656 of *Communications in Computer and Information Science*, Cham. Springer.

Sona Binu, Jismi Jose, Fathima Shimna K V, Alino Luke Hans, Reni K. Cherian, Starlet Ben Alex, Priyanka Srivastava, and Chiranjeevi Yarra. 2024. Language-agnostic analysis of speech depression detection. *Preprint*, arXiv:2409.14769.

Klara Daly and Oluwafemi Olukoya. 2025. Depression detection in read and spontaneous speech: A multimodal approach for lesser-resourced languages. *Biomedical Signal Processing and Control*, 108:107959.

S. Gupta, G. Agarwal, S. Agarwal, et al. 2024. Depression detection using cascaded attention based deep learning framework using speech data. *Multimedia Tools and Applications*, 83:66135–66173.

Spencer L James, Degu Abate, Kalkidan Hassen Abate, Solomon M Abay, Cristiana Abbafati, Nooshin Abbasi, Hedayat Abbastabar, Foad Abd-Allah, Jemal Abdela, Ahmed Abdelalim, et al. 2018. Global, regional, and national incidence, prevalence, and years lived with disability for 354 diseases and injuries for 195 countries and territories, 1990–2017: a systematic analysis for the global burden of disease study 2017. *The lancet*, 392(10159):1789–1858.

N. Janardhan and N. Kumaresh. 2022. Improving depression prediction accuracy using fisher score-based feature selection and dynamic ensemble selection approach based on acoustic features of speech. *Traitement du Signal*, 39(1):87–107.

A. Kim, E. Jang, S. Lee, K. Choi, J. Park, and H. Shin. 2023. Automatic depression detection using smartphone-based text-dependent speech signals: Deep convolutional neural network approach. *Journal of Medical Internet Research*, 25:e34474.

A. Kritika, S. Meenakshy, Arya Palackal Shijish, Riya Rajeev, and G. Jyothish Lal. 2025. DraviMood: Speech-Based Depression Classification in Dravidian Languages Using Feature Fusion and Deep Learning. In *Proceedings of the Fourth International Conference on Speech and Language Technologies for Low-Resource Languages (SPELLL 2025)*.

Jyothish G. Lal, B. Premjith, Bharathi Raja Chakravarthi, Saranya Rajiakodi, Thenmozhi Durairaj, and Prasanna Kumar Kumaresan. 2026. Shared task on depression detection from malayalam and tamil speech data. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Anjali Mathew, Raniya, Harsha Sanjan, Amjith S B, Reni K Cherian, Starlet Ben Alex, Priyanka Srivastava, and Chiranjeevi Yarra. 2024. Instant-emdb: A multi-model spontaneous english and malayalam speech corpora for depression detection. In *2024 27th Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA)*, pages 1–6.

B. Premjith, G. Jyothish, V. Sowmya, and B. Bharathi. 2024. Findings of the Shared Task on Multimodal Social Media Data Analysis in Dravidian Languages (MSMDA-DL)@DravidianLangTech 2024. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 56–61.

Khosro Rezaee. 2026. Depression detection from speech data using deep learning–based optimized

temporal–frequency–channel attention with interpretable acoustic–prosodic mapping. *Journal of Affective Disorders*, 399:121077.

Felix Teufel, Aastha Aggarwal, Lydia Chwastiak, Vikram Patel, and Mohammed K. Ali. 2024. Depression diagnosis, treatment, and remission among adults in india. *JAMA Psychiatry*, 81(12):1265–1269.

Faming Yin, Jing Du, Xinzhou Xu, and Li Zhao. 2023. Depression detection in speech using transformer and parallel convolutional neural networks. *Electronics*, 12(2).

Xu Zhang, Xiangcheng Zhang, Weisi Chen, Chenlong Li, and Chengyuan Yu. 2024. Improving speech depression detection using transfer learning with wav2vec 2.0 in low-resource environments. *Scientific Reports*, 14:9543.

A Appendix

A.1 Bootstrap-Based Robustness Analysis

To assess the stability and statistical reliability of model performance, we applied bootstrap resampling on test set predictions. We estimated the distribution of macro-F1 scores over 1000 resamples and reported the 95% confidence interval (CI). This allowed evaluation beyond point estimates, capturing variability across different sampled subsets of the data.

A.1.1 Tamil Test Set

Model	F1 (%)	CI (L)	CI (U)
MFCC–CNN	78.00	72.00	83.00
Wav2Vec2	47.92	40.77	56.36
Validation-tuned Ensemble	**94.38**	**90.61**	**97.50**

Table 5: Bootstrap Macro-F1 with 95% CI (Tamil)

The Tamil results revealed clear differences in both performance and robustness. MFCC–CNN achieved moderate performance with relatively stable confidence bounds, whereas Wav2Vec2 exhibited low macro-F1 and high variability, indicating weak generalization. In contrast, the proposed ensemble substantially improved both accuracy and stability, achieving the highest score with a comparatively narrow CI.

A.1.2 Malayalam Test Set

In contrast to Tamil, all models achieved consistently high performance with narrow confidence intervals, suggesting stable behavior across resamples. The limited spread indicated low dataset variability, which may inflate performance estimates. Consequently, the ensemble provided only marginal improvement, while Wav2Vec2 showed slightly higher variability, reflecting its sensitivity to data distribution.

Model	F1 (%)	CI (L)	CI (U)
MFCC–MLP	98.51	97.48	99.00
Wav2Vec2	97.04	95.07	98.69
Refined Wav2Vec2	**99.01**	**95.90**	**99.80**

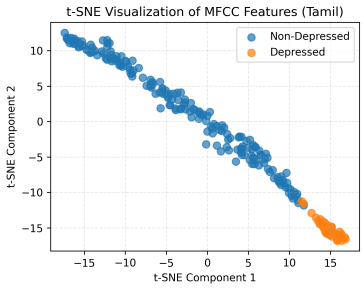
Table 6: Bootstrap Macro-F1 with 95% CI (Malayalam)

A.2 Representation-Level Analysis via t-SNE

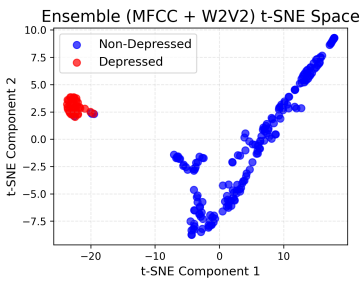
In this section, we conducted a representation-level analysis using t-SNE visualization. We extracted embeddings from intermediate layers prior to classification and applied t-SNE for dimensionality reduction to project them into a two-dimensional space. This allowed us to examine how effectively each model separates the Depressed and Non-Depressed classes in the learned feature space. Since the final systems for both languages were primarily driven by Wav2Vec2 representations, we showed only two t-SNE visualizations: one for the MFCC-based model and one for the final Wav2Vec2-driven system.

A.2.1 Tamil Dataset

We visualized the embeddings using t-SNE, as shown in Fig. 3



(a) MFCC embeddings



(b) Embeddings of Validation-tuned Ensemble

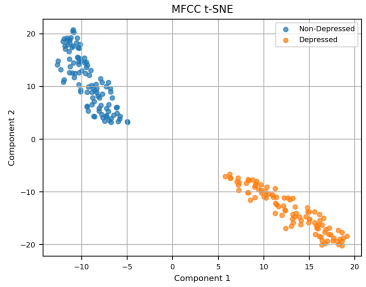
Figure 3: t-SNE visualization for the Tamil dataset.

While MFCC captured basic spectral features, it

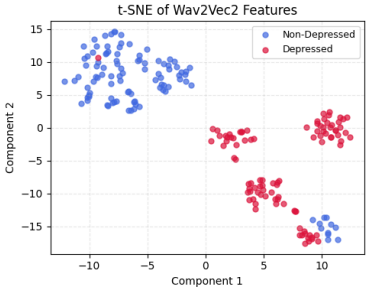
struggled with clear class separation, as evidenced by the linear distribution and overlap. In contrast, the Ensemble plot showed a much clearer separation between the classes. Wav2Vec2, through its self-supervised learning, captured high-level semantic and phonetic features from the Tamil dataset, allowing the model to isolate the Depressed samples into a dense, distinct cluster.

A.2.2 Malayalam Dataset

t-SNE visualizations of the Malayalam embeddings are shown in Fig. 4.



(a) MFCC Embeddings



(b) Refined Wav2Vec2 Embeddings

Figure 4: t-SNE visualization for Malayalam.

The t-SNE plots for MFCC and Wav2Vec2 confirmed high linear separability. While MFCC formed two isolated clusters, Wav2Vec2 displayed granular sub-clusters because it is a self-supervised deep neural network capturing complex, hierarchical acoustic patterns. The minor overlaps in Wav2Vec2 occurred due to inter-class similarity, where the model detected nuanced speech cues (like low energy) shared by both classes. Despite these overlaps, the transformer-based architecture provided a richer representation, ensuring reliable class discrimination.

VITECH@DravidianLangTech2026: Prompting and LoRA Adaptation for Tamil Abusive Language Detection - A Comparative Study of Open LLMs

Triambiga Krubhakaran **Senthil Kumar B**

Kaviya Nagarajan **Balaji N**

Velammal Institute of Technology
Chennai, Tamil Nadu, INDIA
{it24079, senthilkumar.it, it24036, balajin}@velammalitech.edu.in

Abstract

The detection of abusive Tamil text using large language models (LLMs) has received relatively little attention compared to BERT variations. We empirically evaluated four families of open-weight LLMs —Gemma, LLaMA, Qwen, and DeepSeek-Distilled— on the Tamil dataset provided by the shared task. The models are assessed under two in-context learning settings (zero-shot and few-shot) and a parameter-efficient fine-tuning approach using LoRA, with model sizes of approximately 2B and 8B parameters. Experimental results show that 8B models consistently outperform their 2B counterparts, indicating the benefit of increased model capacity. Among the adaptation techniques, LoRA fine-tuning significantly outperforms both zero-shot and few-shot prompting. Across all evaluated settings, Google's Gemma-2-9B model with LoRA fine-tuning achieved the best performance compared to the other model families and our test result was ranked 12th among all 22 submissions with the 0.7959 f1-score.

1 Introduction

The proliferation of user-generated content on social media has intensified the spread of abusive and offensive language online. Automatic abusive language detection has therefore become a critical research problem in natural language processing (NLP), particularly for content moderation and online safety applications (Schmidt and Wiegand, 2017; Fortuna and Nunes, 2018). While substantial progress has been made for high-resource languages such as English (Zampieri et al., 2019), comparatively limited work has focused on low-resource and morphologically rich languages like Tamil.

Tamil presents several linguistic and computational challenges for abusive language detection. Social media text frequently contains spelling variations, dialectal forms, informal expressions, transliteration into Latin script, and Tamil–English code-mixing. These characteristics complicate lexical matching and contextual modeling. Furthermore, the scarcity of large-scale annotated corpora restricts the effectiveness of fully supervised approaches (Jauhiainen et al., 2021). Recent advances in large language models (LLMs) have demonstrated strong zero-shot and few-shot generalization across diverse NLP tasks (Brown et al., 2020; Touvron et al., 2023). Open-weight models have further enabled task adaptation through parameter-efficient fine-tuning methods such as Low-Rank Adaptation (LoRA) (Hu et al., 2022). However, their effectiveness in detecting abusive text in low-resource areas such as Tamil text remains underexplored.

We systematically evaluate four open-weight LLM families: Gemma, Llama, DeepSeek Distilled Qwen, and Qwen. For each family, we experiment with 2B and 8B parameter variants to analyze the impact of model scale. We assess three adaptation paradigms: 1.Zero-shot prompting, 2.Few-shot in-context learning and 3.Parameter-efficient fine-tuning using LoRA. This experimental design enables a controlled comparison between prompt-based inference and supervised adaptation strategies.

Our study aims to address the following research questions (**RQ**):

- **RQ1:** How effective are open-weight LLMs in zero-shot and few-shot settings for Tamil abusive text detection?
- **RQ2:** What is the impact of model scale (2B vs 8B) on classification performance?
- **RQ3:** To what extent does LoRA-based fine-tuning improve performance over prompt-only approaches?

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 436–441
July 3, 2026 ©2026 Association for Computational Linguistics

2 Related Work

A recent UN Women-commissioned survey [1] of women in the public sphere reveals that around 70% of women participants who work in the fields of human rights, activism, and/or journalism said that they had experienced online violence in the course of their work. This emphasizes the need to detect abusive text on social networks.

Early shared tasks on abusive Tamil content were organized at the DravidianLangTech workshop series. In DravidianLangTech-ACL 2022 (Priyadharshini et al., 2022), the task focused on classifying abusive comments in Tamil and Tamil–English code-mixed text, the participants employing machine learning, deep learning, and transformer methods such as mBERT and MuRIL BERT (Rajalakshmi et al., 2022). Subsequent shared tasks expanded the scope to Tamil and Telugu abusive comment detection (Priyadharshini et al., 2023), with competitive systems leveraging transformer-based architectures such as mBERT and XLM-RoBERTa achieving the highest performance (Hegde et al., 2023).

More recently, the shared task at NAACL 2025 under the DravidianLangTech workshop (Rajiakodi et al., 2025) focused on abusive Tamil and Malayalam text targeting women on social media. The findings emphasized the effectiveness of the multilingual pre-trained model IndicBERT-v2 over the other BERT variants (Hanif and Rahman, 2025). Despite these advances, prior work has largely focused on fully supervised fine-tuning of encoder-based transformer models. Systematic investigations of large language models (LLMs) based on decoder-only for Tamil abusive text detection remain limited.

Our work addresses this gap by conducting a controlled evaluation of multiple open-weight LLM families on different parameter scales (2B and 8B) under zero-shot, few-shot, and LoRA-based adaptation paradigms within a shared task setting. This analysis offers new insights into the trade-offs between model scale, adaptation strategy, and performance in Dravidian NLP contexts.

3 Dataset Statistics

The dataset used in this study was released as part of the shared task by Sivagnanam et al. (2026). It consists of manually annotated Tamil sentences labeled as either Abusive or Non-Abusive. The training split contains 3,652 sentences, while the test split comprises 913 sentences. The training data is relatively balanced across classes, with 1,769 abusive instances and 1,883 non-abusive instances. This near-balanced class distribution reduces bias toward majority-class predictions and supports reliable macro-level evaluation.

In terms of sentence length, the dataset exhibits notable variability. In the training set, sentence length ranges from a minimum of 3 words to a maximum of 383 words, with an average length of 14.48 words per sentence. Similarly, the test set ranges from 6 to 158 words per sentence, with an average of 13.95 words per sentence. Although the average sentence length is relatively short, the presence of long outlier instances introduces modeling challenges, particularly for prompt-based methods with context length constraints.

4 Experiment

We evaluated two model scales - approximately 2B and 8B parameters - across four open-weight large language model (LLM) families: Gemma, Llama, DeepSeek-R1-Distill-Qwen, and Qwen3. The given training dataset of 3652 sentences is split (80:20) into training and validation of 2921 and 731 sentences respectively. The best model across all settings was used to submit our result for the given test data.

4.1 Model Description

For our experiments, we utilized eight decoder-only transformer models on two parameter scales (1–2B and 7–9B), all obtained from Hugging Face and executed in the Google Colab Pro environment. The smaller models (1–2B), hereafter called 2B models, were used for lightweight evaluation and adaptation, while the higher-capacity models (7–9B), hereafter called 8B models, enabled more expressive contextual modeling in identical experimental settings.

In both 2B and 8B model sizes, we used the Gemma series (Gemma-2B[2], Gemma-2-9B[3]) developed by Google DeepMind; Qwen series (Qwen3-1.7B [4] & Qwen2.5-7B [5]) from Alibaba Cloud, a multilingual foundation model optimized

[1] http:https://www.unesco.org/en/articles/global-survey-reveals-rising-violence-against\-women-journalists

[2] https://huggingface.co/google/gemma-2b

[3] https://huggingface.co/google/gemma-2-9b

[4] https://huggingface.co/Qwen/Qwen3-1.7B

[5] https://huggingface.co/Qwen/Qwen2.5-7B

for reasoning; Llama-Tamil (Llama-Tamil-1B [6] & Tamil-Llama-7b[7]), a Tamil-adapted variant of Llama originally developed by Meta AI; and DeepSeek-R1-Distill-Qwen series (Distill-Qwen-1.5B [8] & Distill-Qwen-7B[9]) from DeepSeek, a distilled model designed to retain reasoning capabilities in a smaller architecture.

These models were evaluated under zero-shot, few-shot, and LoRA settings for the detection of abusive Tamil text.

4.2 Adaptation Strategies

Each model was evaluated under three adaptation paradigms:

4.2.1 Zero-Shot Prompting

The models were prompted with a structured instruction-based template as shown in the Appendix.A, specifying the classification labels (abusive or non-abusive). Instead of updating model parameters, the task is described entirely through natural language instructions within the prompt. The model leverages the knowledge acquired during pretraining to map the input text to one of the predefined class labels.

4.2.2 Few-Shot Prompting

The large language model (LLM) learns the task pattern from in-context examples and applies it to a new, unseen instance. This technique leverages the model's in-context learning capability. In the few-shot setting, 5 labeled examples from each class were included in the prompt prior to the test instance. Examples were selected to ensure a balanced representation of both classes. The prompt structure used in few-shot learning is shown in the Appendix.A

4.2.3 LoRA-based Fine-tuning

For supervised adaptation, we used Low-Rank Adaptation (LoRA) (Hu et al., 2022). LoRA layers were injected into the attention projection matrices while freezing the base model parameters. This approach enables parameter-efficient fine-tuning, which is particularly suitable for moderate dataset sizes in low-resource languages.

[6] https://huggingface.co/Jesmma/LLaMA-Tamil_1B

[7] https://huggingface.co/abhinand/tamil-llama-7b-base-v0.1

[8] https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B

[9] https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B

2B LLM	zero-shot	few-shot
Gemma-2b	**56.05**	33.86
Llama-Tamil-1B	**49.27**	48.50
DeepSeek-R1-Distill-Qwen-1.5B	49.72	**51.15**
Qwen3-1.7B	12.07	**58.59**

Table 1: Effectiveness of 2B LLMs in zero-shot & few-shot settings. **Bold** indicates the better f1-score among the prompting

8B LLM	zero-shot	few-shot
Gemma-2-9B	**64.06**	58.17
Tamil-Llama-7B	**65.28**	65.19
DeepSeek-R1-Distill-Qwen-7B	**61.77**	15.10
Qwen2.5-7B	58.71	**65.51**

Table 2: Effectiveness of 8B LLMs in zero-shot & few-shot settings. **Bold** represents the better f1-score among the prompting

4.3 Implementation

All experiments were implemented using the HuggingFace Transformers library, with the PEFT framework for LoRA-based parameter-efficient fine-tuning. The NVIDIA T4 GPU (16GB VRAM) available in the Google Colab free tier serves as an effective computational tool for 2B parameter models. The 8B parameter models was trained on a NVIDIA A100 GPU with 40GB VRAM.

Implemented LoRA for both 2B and 8B parameter models to compare with the performance of the prompting techniques. For both the 2B and 8B models, LoRA was configured as given in the Appendix.B. A higher rank in 2B allows the adapter to capture more complex features that the model might otherwise miss. The 8B models use a lower rank to maintain memory efficiency without overparameterizing already large representations. For 2B models, LoRA is applied only to attention projections (q, k, v, o), focusing on contextual representation learning. For 8B models, adaptation extends to MLP layers, enabling a deeper task-specific transformation. Both 2B and 8B models were adapted by fine-tuning the hyper parameters in LoRA as given in the Appendix.B.

5 Results

The four types of open-weight LLMs at two sizes -2B & 8B- are empirically evaluated using the validation dataset and compared for their effectiveness

2B/8B LLM	Prompt	LoRA	Diff. (△)	% increase
Gemma-2B	56.05	78.75	22.7	40%
Llama-Tamil-1B	49.27	58.04	8.77	18%
Distill-Qwen-1.5B	51.15	73.04	21.89	43%
Qwen3-1.7B	58.59	77.48	18.89	43%
Gemma-2-9B	64.06	81.49	17.43	27%
Tamil-Llama-7B	65.28	80.32	15.04	23%
Distill-Qwen-7B	61.77	69.77	8	13%
Qwen2.5-7B	65.51	76.1	10.59	16%

Table 3: Improvement in the performance of LoRA fine-tuning over prompt-only approach. The best f1-score among zero/few-shot in each model is considered as the prompt score.

LoRA Adaptation		
LLM	**2B**	**8B**
Gemma	78.75	**81.49**
Llama	58.04	**80.32**
DeepSeek-R1-Distill-Qwen	**73.04**	69.77
Qwen	**77.48**	76.10

Table 4: Effectiveness of LoRA adaptation in 2B & 8B LLMs. Bold indicates the better f1-score among the model sizes.

over the three adaptation techniques: zero-shot, few-shot and LoRA. Their effectiveness can be analyzed by answering the research questions (**ARQ**):

- **ARQ1:** From Table.1 & 2, it is evident that Zero-shot works better when the model already has strong multilingual semantic grounding, whereas few-shot helps when the model needs explicit label anchoring or format conditioning.

- **ARQ2:** By comparing the f1-scores in Table.1, 2 across the 2B and 8B LLMs for the prompting techniques, generally 8B size LLMs perform better in zero-shot and few-shot techniques, except for DeepSeek-Distill model. For LoRA fine-tuning, the behavior differs between model families (Table.4). In the case of Gemma and Llama-based models, the 8B variants benefit more from LoRA adaptation. In contrast, DeepSeek-R1-Distill-Qwen and Qwen models perform better with 2B size LLMs. This behavior may be attributed to the relatively small size of the training dataset, where smaller models generalize more effectively and are less prone to overfitting during LoRA adaptation. Furthermore, abusive Tamil text classification is a domain-specific task that may not require the additional representational capacity of larger models. The distilled architecture of DeepSeek-Distill-Qwen-2B may also contribute to improved task efficiency and discriminative learning for classification-oriented objectives.

- **ARQ3:** The percentage increase in performance by LoRA fine-tuning over prompt-only approach implies that LoRA fine-tuning is better than the prompt techniques as shown in Table.3. The LoRA adaptation improve the performance by 36% and 20% approximately for 2B and 8B LLMs respectively, over prompt-only approaches.

6 Conclusion

The open-weight LLMs of two sizes - 2B & 8B - was empirically evaluated through three adaptation techniques. Among the three, LoRA fine-tuning performs better than the two prompting techniques. Although the percentage improvement by LoRA is greater in 2B size LLMs due to poor performance in prompting techniques, 8B LLMs such as Gemma-2-9B and Tamil-Llama-7B achieved the highest f1-score with 81.49% and 80.32% respectively, during the validation. The predictions for the test data are inferred from the top three LoRA fine-tuned LLM models, Gemma-2-9B, Tamil-Llama-7B and Qwen3-1.7B. The system scored a 12th rank among the 22 submissions.

Overall, the above study on open-weight LLM in 2B and 8B sizes infers that the LLM which is multilingual, larger in model size, and fine-tuning performs better in low-resource language.

7 Limitations

The study uses a relatively small dataset of approximately 3,600 samples. Large language models, particularly 8B-scale models, generally require substantially larger datasets for stable adaptation and generalization. Although four LLM families were evaluated, the study focuses primarily on models in the 2B and 8B parameter range. Additionally, the evaluation is restricted to four LLM families and a single shared-task setting focused on abusive Tamil text classification. The study also does not investigate tokenizer behavior, dialectal variation, or advanced prompting strategies, all of which can influence performance in low-resource multilingual contexts.

8 Ethical Considerations

This work uses the publicly released dataset from the shared task organizers solely for research purposes. Since the dataset contains offensive and harmful content targeting women, systems can produce false positives or false negatives, especially in low-resource languages such as Tamil, human oversight remains important in real-world deployment. Although this work evaluates model performance on a downstream classification task, it does not provide a comprehensive fairness analysis.

Limited AI-assisted writing tools were used to improve the grammatical clarity and presentation of the manuscript. All experimental design, implementation, analysis, and interpretation of results were conducted and verified by the authors. The authors carefully reviewed and validated all AI-assisted content to ensure accuracy, originality, and compliance with academic integrity standards.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Comput. Surv.*, 51(4).

Tareque Md Hanif and Md Rashadur Rahman. 2025. CUET_Agile@DravidianLangTech 2025: Fine-tuning transformers for detecting abusive text targeting women from Tamil and Malayalam texts. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 315–319, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Asha Hegde, Kavya G, Sharal Coelho, and Hosahalli Lakshmaiah Shashirekha. 2023. MUCS@DravidianLangTech2023: Leveraging learning models to identify abusive comments in code-mixed Dravidian languages. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, pages 266–274, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Tommi Jauhiainen, Tharindu Ranasinghe, and Marcos Zampieri. 2021. Comparing approaches to Dravidian language identification. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 120–127, Kiyv, Ukraine. Association for Computational Linguistics.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, Subalalitha Chinnaudayar Navaneethakrishnan, Thenmozhi Durairaj, Malliga Subramanian, Kogilavani Shanmugavadivel, Siddhanth U Hegde, and Prasanna Kumar Kumaresan. 2022. Overview of abusive comment detection in Tamil-ACL 2022. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 292–298, Dublin, Ireland. Association for Computational Linguistics.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, Malliga Subramanian, Subalalitha Chinnaudayar Navaneethakrishnan, Kogilavani Shanmugavadivel, Premjith B, Abirami Murugappan, Prasanna Kumar Kumaresan, Karnati Sai Prashanth, Mangamuru Sai Rishith Reddy, and Janakiram Chandu. 2023. Overview of shared-task on abusive comment detection in Tamil and Telugu. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, pages 80–87, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Ratnavel Rajalakshmi, Ankita Duraphe, and Antonette Shibani. 2022. DLRG@DravidianLangTech-ACL2022: Abusive comment detection in Tamil using multilingual transformer models. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 207–213, Dublin, Ireland. Association for Computational Linguistics.

Saranya Rajiakodi, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, Ruba Priyad-

Prompt:
prompt = f"""Task: Classify the following text as either "abusive" or "non-abusive". Text: {text} Classification:"""

Table 5: Zero-shot

harshini, Rajameenakshi J, Kathiravan P, Rahul Ponnusamy, Bhuvaneswari Sivagnanam, Paul Buitelaar, Bhavanimeena K, Jananayagam V, and Kishore Kumar Ponnusamy. 2025. Findings of the shared task on abusive Tamil and Malayalam text targeting women on social media: DravidianLangTech@NAACL 2025. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 671–681, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Bhuvaneswari Sivagnanam, Kathiravan Pannerselvam, Jananayagan V, Charmathi Rajkumar, Ramesh Kannan R, Ratnavel Rajalakshmi, Shunmuga Priya Muthusamy Chinnan, Saranya Rajiakodi, and Bharathi Raja Chakravarthi. 2026. From Comments to Harm: A Findings Report on Abusive Tamil Text Targeting Women on Social Media. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

9 Appendix

A Prompts

The prompts used in zero-shot and few-shot learning for 2B and 8B size LLMs are shown in Table.5, 6.

Prompt:
prompt=f"""Classify Tamil text as "abusive" or "non-abusive". Respond with ONLY one word. {FEW_SHOT_EXAMPLES} Now classify this text. Respond with only "abusive" or "non-abusive": Text:{text} Classification:"""

Table 6: Few-shot

Config	**2B**	**8B**
Rank - r	16	8
Alpha - α	32	16
Scaling factor	2*r	2*r
Target	attention	attention+MLP

Table 7: LoRA Configuration

B LoRA Configuration & Hyper parameters

The LoRA configuration during LLM adaptation and the corresponding fine-tuning hyperparameters in Table.7, 8 respectively.

Parameter	**2B**	**8B**
Batch size	32	32
Learning rate	2e-4	1e-4
Weight decay	0.01	0.01
Warm-up ratio	0.1	0.1
Optimizer	adamw_torch	paged_adamw

Table 8: LoRA Hyper parameters

Wave2Word@DravidianLangTech 2026: WhisTam: A unified framework for dialect based Tamil speech recognition and classification

Ruwad Naswan[1], Shadab Tanjeed Ahmad[2],
[1]Bangladesh University of Engineering and Technology, [2]Islamic University of Technology,
Correspondence: ruwad45678@gmail.com

Abstract

While Automatic Speech Recognition (ASR) systems have shown impressive performance in languages having sufficient annotated speech data like English, their performance is still limited for low-resource, dialect rich languages like Tamil. Tamil poses further challenges because of its extremely high regional variation in dialects that manifest in varying vocabulary, pronunciations, and even syntactic structures. To address these challenges, we present a unified framework WhisTam based on the Whisper medium model, which performs speech transcription and dialect classification jointly within a single system. Our method is evaluated against speech samples from four regional dialects and achieves a macro F1-score of 0.53 and a Word Error Rate (WER) of 0.55 for dialect classification and transcription respectively, ranking 2nd in the dialect classification task and 3rd in the transcription task in the DravidianLangTech@ACL 2026 shared task on Dialect-based Speech Recognition and Classification in Tamil. These findings emphasize the challenges in dialectal Tamil ASR as well as the promise of multi-task learning for low-resource languages. Our implementation is publicly available at: https://github.com/rwd51/DravidianLangTech-Wave2Word.

1 Introduction

Recent advances in deep learning and transformer-based architectures have greatly improved ASR for well-resourced languages (Radford et al., 2022; Baevski et al., 2020; Hinton et al., 2012). However, progress remains uneven across languages, particularly for low-resource, dialect-rich languages where annotated speech data is scarce (Besacier et al., 2014; Pratap et al., 2020). Insufficient transcribed data combined with phonetic and acoustic variability leads to substantially higher error rates for such languages (Besacier et al., 2014). Multilingual training, transfer learning, and parameter-efficient adaptation of pretrained models have emerged as promising directions to bridge this gap (Conneau et al., 2020).

Tamil, a major Dravidian language spoken by over 80 million people (Nanmalar et al., 2024a), remains relatively underexplored in ASR research (Pratap et al., 2020). Its high regional dialectal variation in pronunciation, vocabulary, and syntax poses a significant challenge, especially when ASR systems are trained on limited or standardized corpora.

To address this, we propose WhisTam, a unified framework that jointly performs speech transcription and dialect classification for Tamil. Built on a pre-trained Whisper model (Radford et al., 2022), WhisTam introduces a lightweight classifier adapter on the encoder to simultaneously identify the dialect and generate transcriptions across four regional groups: northern, southern, western, and central. Our experiments yield a macro F1-score of 0.53 for dialect classification and a WER of 0.55 for transcription, highlighting both the difficulty of dialectal Tamil ASR and the promise of multi-task learning for low-resource settings.

2 Related Works

Early Dravidian speech processing relied on classical features like MFCC combined with GMMs, HMMs, and SVMs for language and dialect classification (Ismail and Singh, 2017; Koolagudi et al., 2012), achieving around 88% accuracy across Tamil, Telugu, Kannada, and Malayalam (Abdul and Al-Talabani, 2022; Koolagudi et al., 2012). Deep learning approaches have since pushed these results further: CNN-based spectrogram representations reached over 98% on Indic speech datasets (Godbole et al., 2020), and MFCC with handcrafted feature combinations achieved F1 scores above 0.99 for Tamil dialect identification (Nan-

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 442–446
July 3, 2026 ©2026 Association for Computational Linguistics

malar et al., 2019, 2022, 2024b). Multilingual ASR training and multitask phone mapping have also shown gains in low-resource Dravidian settings by leveraging cross-language acoustic similarities (Madhavaraj and Ganesan, 2022).

Despite this progress, dialectal variation, speaker diversity, and limited labeled data continue to hinder robust ASR for most Dravidian languages (Das and Bhattacharjee, 2024; Ahmad Dar and Pushparaj, 2025), motivating frameworks that can generalize across dialects and low-resource conditions.

3 Methodology

3.1 Problem Statement

This work is conducted as part of the DravidianLangTech 2026 (Bharathi et al., 2025, 2026) shared task at ACL, which focuses on advancing speech technologies for dialectal Tamil.

The shared task consists of two subtasks designed to evaluate models on dialect-aware speech understanding.

3.1.1 Subtask 1: Speech-Based Dialect Classification

The goal of this subtask is to determine the dialect of a provided Tamil speech recording and categorize each given audio sample into one of four dialect categories: *Northern*, *Southern*, *Western*, or *Central*.

3.1.2 Subtask 2: Dialectal Tamil Automatic Speech Recognition

The second subtask focuses on automatic speech recognition for dialectal Tamil. Participants are required to generate Tamil text transcriptions from speech recordings containing dialectal variation.

3.2 Architecture Overview

Our framework builds upon a Tamil-adapted version of the Whisper model (Radford et al., 2022). Specifically, we initialize from vasista22/whisper-tamil-medium, a publicly available model fine-tuned on multiple Tamil ASR corpora. Whisper is a model that transforms input speech into log-Mel spectrogram features. These features are fed through a transformer encoder to create contextual acoustic representations, which are then used by a transformer decoder to produce autoregressive text. Our approach retains the core encoder and decoder architecture but introduces lightweight extensions that enables dialect-aware learning.

Regional Adapter. To incorporate dialect awareness, we introduce a *Regional Adapter* layered on top of the encoder. A small embedding layer maps each dialect label (Northern, Southern, Western, Central) into a low-dimensional vector. This vector is projected to the encoder hidden dimension using a linear layer and normalized via LayerNorm. During training, the projected embedding is broadcast-added to every timestep of the encoder hidden states:

$$\mathbf{h}'_t = \mathbf{h}_t + \mathbf{e}_{region}$$

This allows the model to condition its representations on dialect-specific characteristics such as phonetic cues and prosody. During inference, the dialect embedding is omitted, and the model operates as a standard Whisper ASR system.

Dialect Classification Head. To explicitly learn dialect-discriminative features, we attach an auxiliary classification head. Encoder hidden states are aggregated using an attention-based pooling mechanism, which assigns higher weights to frames informative for dialect recognition. The pooled representation is passed through dropout and a linear classifier to predict the dialect category. Attention pooling had been used over simple mean or max pooling as dialect markers are not uniformly distributed across time.

Multi-task Learning. The model is trained with a multi-task objective combining ASR and dialect classification. Let $\mathcal{L}_{ASR}$ denote the standard Whisper cross-entropy loss for next-token prediction, and $\mathcal{L}_{dialect}$ denote the cross-entropy loss for dialect classification. The total training loss is:

$$\mathcal{L}_{total} = \mathcal{L}_{ASR} + \alpha \cdot \mathcal{L}_{dialect} \quad (1)$$

where α balances the contribution of the dialect classification loss. This multi-task setup encourages the shared encoder to organize its representations around dialect-specific patterns while maintaining strong transcription performance.

Overall, the Regional Adapter introduces a small number of additional parameters relative to the base Whisper model, enabling dialect-aware ASR with minimal architectural modification.

4 Experiments

4.1 Dataset and Preprocessing

Dataset. The Tamil Dialect Speech Dataset (Bharathi et al., 2025, 2026) covers four major

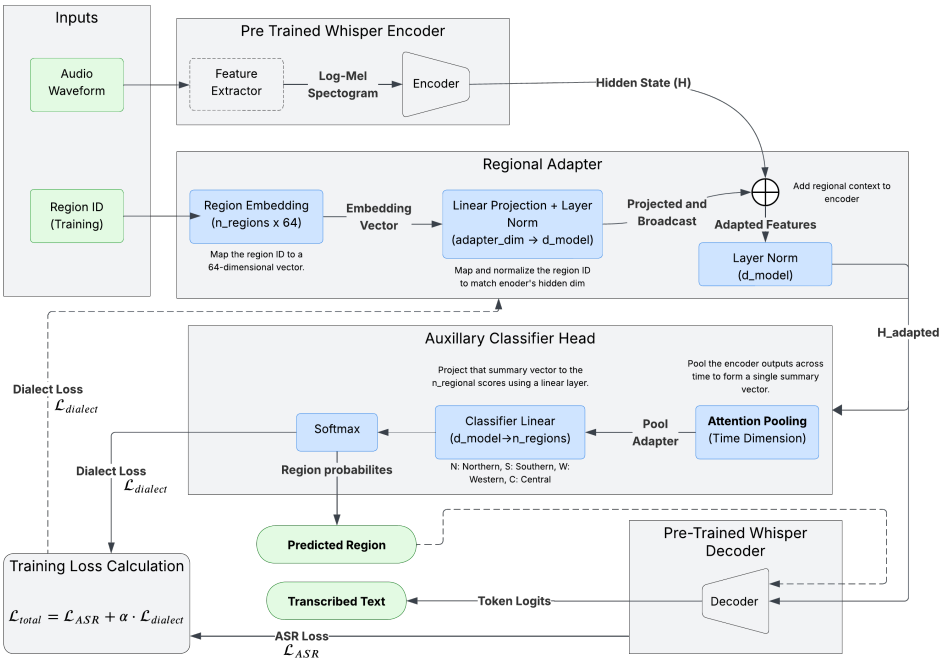


Figure 1: Classifier adapter architecture used in the regional Whisper model.

dialect regions of Tamil Nadu: Northern, Southern, Western, and Central. It contains both spontaneous and read speech recorded by male and female speakers across diverse age groups in natural acoustic environments at 16 kHz. The training set consists of 9.22 hours of manually transcribed speech, and 2.05 hours are released as test set for evaluation. The 9.22-hour training corpus is partitioned into a 90/10 train/validation split using stratified sampling.

Preprocessing. Text normalization is applied to transcriptions, including Unicode NFC normalization, removal of zero-width characters, Tamil visarga normalization, punctuation removal, English lowercasing, and whitespace normalization. Audio is converted to log-mel spectrograms using the Whisper feature extractor, zero-padded or truncated to a fixed length.

During training, audio-level augmentations are applied via curriculum learning, including time-stretching, pitch shifting, additive Gaussian or pink noise, volume shifts, time shifts, and random cropping. SpecAugment is further applied to spectrograms, masking frequency and time bands to improve robustness to acoustic variability.

4.2 Evaluation

For **dialect classification**, system performance is measured using macro-averaged F1 score across the four dialect classes.

For **dialectal ASR**, performance is measured using Word Error Rate (WER) between predicted transcriptions and reference human transcriptions:

$$WER = \frac{S + D + I}{N}$$

where S, D, and I are the numbers of substitutions, deletions, and insertions, respectively, and N is the total number of words in the reference.

4.3 Results

We report results for both tasks on the stratified validation split from the training set, as well as on the hidden test set used in the shared task competition.

4.3.1 Automatic Speech Recognition (ASR)

Table 1 shows per-dialect WER on the validation set and the overall WER (mean across dialects) on validation as well as the overall WER on the hidden test set. Lower WER indicates better transcription quality.

Figure 2: t-SNE visualization of encoder embeddings on the validation set. Left: pretrained Whisper encoder (mean-pooled). Right: trained regional adapter (attention-pooled). The trained model produces clearly separated dialect clusters.

Dialect / Set	**Validation WER (%)**	**Hidden Test WER (%)**
Central Dialect	30.23	–
Northern Dialect	36.70	–
Southern Dialect	40.08	–
Western Dialect	41.68	–
Mean / Overall	37.17	55.0

Table 1: ASR performance (WER %) for each Tamil dialect on the validation set and overall on the hidden test set.

4.3.2 Dialect Classification

Table 2 shows precision, recall, and F1 for each dialect on the validation set, and the overall macro F1 on the hidden test set. Higher values indicate better classification performance.

Table 2: Dialect classification performance on validation and hidden test set.

Dialect	**Precision**	**Recall**	**F1 Score**	**Hidden Macro F1**
Northern	0.97	1.00	0.98	–
Southern	0.98	1.00	0.99	–
Western	1.00	0.90	0.95	–
Central	0.94	1.00	0.97	–
Overall	0.97	0.98	0.97	0.53

4.3.3 Embedding Space Analysis

To qualitatively assess whether the regional adapter encourages dialect-discriminative representations, we visualize the attention-pooled encoder outputs using t-SNE on the held-out validation set. As shown in Figure 2, the pretrained Whisper encoder produces embeddings with no visible dialect structure, while the trained regional adapter yields clearly separated clusters corresponding to the four dialect regions. This confirms that the multi-task objective guides the shared encoder to organize its representations around dialect-specific acoustic patterns.

Conclusion

We introduce WhisTam, a framework developed on top of a Tamil-adapted Whisper model incorporating a small regional adapter for Tamil dialect speech recognition and classification. Our method jointly addresses ASR and dialect identification, leveraging multi-task learning to enhance robustness to dialectal variations. Evaluation on stratified validation and hidden test sets show strong performance of our dialect-aware ASR as well as classification task. Despite the limited size and imbalance of the dataset, our proposed method demonstrates the potential of lightweight adaptation for low-resource dialectal ASR.

Ethical Considerations

This work uses publicly available speech data provided by the DravidianLangTech 2026 shared task organizers with appropriate consent procedures. No additional data is collected or redistributed. We acknowledge that dialect identification systems could potentially be misused for speaker profiling; our intended use is solely to improve speech technology accessibility for underrepresented Tamil

dialect communities.

Limitations

Our dataset is relatively small (9.22 hours) and unevenly distributed across dialects, which may limit generalization to underrepresented speakers and regions. Intra-dialect variation such as speaker age, gender, and code-switching remains difficult for the lightweight adapter to fully capture. Future work could extend the regional adapter to model finer-grained dialectal characteristics and improve cross-dialect ASR performance.

References

Zrar Khald Abdul and Abdulbasit K. Al-Talabani. 2022. Mel frequency cepstral coefficient and its applications: A review. *IEEE Access*, 10:122136–122158.

Muzaffar Ahmad Dar and Jagalingam Pushparaj. 2025. Machine learning and deep learning approaches for accent recognition: A review. *IEEE Access*, 13:51527–51550.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.

Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz. 2014. Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, 56:85–100.

B. Bharathi, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, S. Saranya, and S. Suhasini. 2026. Findings in tamil dialect speech recognition and classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

B. Bharathi, S. Saranya, P. Vijayalakshmi, and T. Nagarajan. 2025. Multi-dialect speech corpus creation for enhancing tamil automatic speech recognition. *Circuits, Systems, and Signal Processing*, pages 1–19.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2020. Unsupervised cross-lingual representation learning for speech recognition. *Preprint*, arXiv:2006.13979.

Hem Das and Utpal Bhattacharjee. 2024. Assamese dialect identification using static and dynamic features from vowel. *Journal of Advances in Information Technology*, 15:306–321.

Shubham Godbole, Vaishnavi Jadhav, and Gajanan K. Birajdar. 2020. Indian language identification using deep learning. *ITM Web of Conferences*.

Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

Tanvira Ismail and L. Joyprakash Singh. 2017. Dialect identification of assamese language using spectral features. *Indian journal of science and technology*, 10:1–7.

Shashidhar G. Koolagudi, Deepika Rastogi, and K. S. Sekhara Rao. 2012. Identification of language using mel-frequency cepstral coefficients (mfcc). *Procedia Engineering*, 38:3391–3398.

A. Madhavaraj and Ramakrishnan Angarai Ganesan. 2022. Data and knowledge-driven approaches for multilingual training to improve the performance of speech recognition systems of indian languages. *Preprint*, arXiv:2201.09494.

M. Nanmalar, S. Johanan Joysingh, P. Vijayalakshmi, and T. Nagarajan. 2024a. A feature engineering approach for literary and colloquial tamil speech classification using 1d-cnn. *Speech Commun.*, 173:103254.

M. Nanmalar, S. Johanan Joysingh, P. Vijayalakshmi, and T. Nagarajan. 2024b. A feature engineering approach for literary and colloquial tamil speech classification using 1d-cnn. *Speech Commun.*, 173:103254.

M. Nanmalar, P. Vijayalakshmi, and T. Nagarajan. 2019. Literary and colloquial dialect identification for tamil using acoustic features. *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, pages 1303–1306.

M. Nanmalar, P. Vijayalakshmi, and T. Nagarajan. 2022. Literary and colloquial tamil dialect identification. *Circuits, Systems, and Signal Processing*, 41:4004 – 4027.

Vineel Pratap, Anuroop Sriram, Paden Tomasello, Awni Hannun, Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert. 2020. Massively multilingual asr: 50 languages, 1 model, 1 billion parameters. *Preprint*, arXiv:2007.03001.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *Preprint*, arXiv:2212.04356.

Wise@DravidianLangTech 2026: Dialect-Aware Tamil Speech Classification and Recognition via Cross-Pipeline Embedding Transfer

Ganesh Sundhar S[1], Hari Krishnan N[1], Gnanasabesan G[1], Suriya KP[1], Jyothish Lal G[1]

[1]Amrita School of Artificial Intelligence, Coimbatore, Amrita Vishwa Vidyapeetham, India

{cb.en.u4aie22017, cb.en.u4aie22020, cb.en.u4aie22018, cb.en.u4aie22164}@cb.students.amrita.edu

g_jyothishlal@cb.amrita.edu

Abstract

This paper presents the **Wise** system for the shared task on dialect-based speech processing in Tamil, addressing two subtasks: **(1) four-way dialect region classification** (Northern, Southern, Western, Central), and **(2) dialectal Tamil ASR**. All audio is preprocessed using loudness normalization followed by neural denoising to ensure consistent audio quality for downstream models. For classification, we experminet different model variants combining multilingual and Tamil-pretrained **Wav2Vec2** backbones with five temporal pooling strategies under frozen and partial fine-tuning settings. Our best configuration i.e. learned attentive pooling with partial fine-tuning and a differentially-trained MLP head achieves a macro F1 of **0.79**, securing **1st place** (0.26-point margin). For ASR, we propose two novel **dialect-conditioned Whisper** architectures (residual injection and cross-attention) that inject dialect embeddings from the trained classifier into the ASR pipeline, and additionally evaluate a vanilla Whisper-Tamil fine-tuned baseline. The best model achieved a WER of **0.90**, securing **8th place**.

Keywords: Dialect Embeddings, Temporal Pooling, Frozen Backbone, Residual & Cross Attention Injection, Wav2Vec2, Whisper

1 Introduction

Tamil is one of the oldest classical languages in the world, spoken by millions across South India, Sri Lanka, and Singapore. Its wide geographic spread has produced rich regional dialects that differ in pronunciation, rhythm, and vocabulary, making it difficult for automatic speech recognition and dialect identification systems to perform well.

The shared task on dialect-based speech recognition and classification in Tamil (Bharathi et al., 2026) curated a dataset to address this limitation and facilitate the training of dialect-aware models (Bharathi et al., 2025). This dataset was used to train the models. The system presented in this paper addresses two subtasks—**(1) dialect identification** and **(2) speech transcription**—and makes three contributions: a preprocessing pipeline combining LUFS normalization and neural denoising, a study of 12 dialect classification variants showing learned attentive pooling performs best, and two dialect-conditioned Whisper architectures that inject dialect embeddings into the ASR encoder.

2 Related Work

Early dialect and accent identification systems relied on acoustic features such as MFCCs combined with statistical models like GMMs or HMMs (Levinson, 1986; Gorin et al., 2014). Deep learning approaches, including LSTMs, CNNs, and RNNs, later achieved significant improvements in speech modeling (Jain et al., 2020; Weninger et al., 2015). More recently, self-supervised models such as Wav2Vec 2.0 (Baevski et al., 2020) and Whisper (Radford et al., 2023) have established new benchmarks for speech processing tasks (S et al., 2025b). Research on hate speech in Dravidian languages (S et al., 2025a) has also progressed considerably using both classical and neural approaches (Sreelakshmi et al., 2024; B et al., 2023). For Tamil specifically, fine-tuned Whisper models have demonstrated strong performance on speech and language tasks (Jairam et al., 2024). Building on these advances, our work studies pooling strategies for dialect classification and integrates dialect embeddings into the ASR pipeline.

3 Dataset

The dataset comprises spontaneous and read speech from native speakers across four regional dialect groups. All recordings are sampled at 16 kHz and captured in natural acoustic environments. The training partition contains 9.22 hours of transcribed speech. Table 1 summarizes the train distribution.

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 447–452
July 3, 2026 ©2026 Association for Computational Linguistics

Dialect	Samples	Duration
Southern	1,427	2h 44m
Northern	1,696	3h 29m
Western	1,126	1h 59m
Central	885	1h 08m
Total	**5,134**	**9h 22m**

Table 1: Train Dataset distribution across dialect groups.

4 Methodology

Our system comprises three tightly integrated stages: audio preprocessing, dialect classification (Subtask 1), and automatic speech recognition (Subtask 2). A key design principle is that the dialect classifier trained in Subtask 1 directly feeds into the ASR system of Subtask 2 through dialect embedding transfer. We describe each component in detail below.

4.1 Audio Preprocessing

The raw audio recordings exhibit considerable variation in recording conditions—including ambient noise, varying volume levels, multi-channel recordings, and DC bias—which can degrade downstream model performance. We design a two-stage preprocessing pipeline to address these issues.

Stage 1 — Audio Standardization. We apply three sequential transformations to each audio file:

1. **Channel Selection**: For multi-channel recordings, we estimate the Signal-to-Noise Ratio (SNR) for each channel. The one with the highest SNR is retained.
2. **DC Offset Removal**: The mean of the waveform is subtracted ($x'(t) = x(t) - \bar{x}$) to eliminate any residual DC bias introduced by recording hardware.
3. **LUFS Normalization**: We normalize loudness to -23 LUFS using the ITU-R BS.1770 standard via the `pyloudnorm` library (Johnson, 2006). Unlike simple peak normalization, LUFS normalization accounts for perceptual loudness, ensuring consistent apparent volume across utterances regardless of their dynamic range.

Stage 2 — Neural Denoising. We apply Facebook's Denoiser DNS64 model (Defossez et al., 2020) for speech enhancement. This model is based on a causal U-Net encoder-decoder architecture that operates directly in the time domain, enabling it to preserve phase information.

4.2 Subtask 1: Dialect Classification

For dialect classification, rather than committing to a single architecture, we conduct a systematic ablation study of 12 model variants to identify the optimal configuration. All variants share a common three-component architecture: (1) a self-supervised speech backbone for feature extraction, (2) a temporal pooling layer to aggregate frame-level representations into a fixed-dimensional utterance embedding, and (3) a multi-layer perceptron (MLP) (Rumelhart et al., 1986) classification head.

4.2.1 Speech Backbone.

We experiment with the original multi-lingual (*facebook/wav2vec2-large-xlsr-53*) model and the Tamil finetuned version (*Harveenchadha/vakyansh-wav2vec2-tamil-tam-250*) as our speech feature extractor. We investigate two fine-tuning strategies:

- **Frozen**: The entire backbone is frozen, and only the pooling and classification layers are trained.
- **Partial Fine-tuning**: The CNN feature encoder and the bottom transformer layers are frozen, while the top N transformer layers are unfrozen and trained jointly with the pooling and classification layers. This allows the model to adapt its higher-level contextual representations to the dialect classification task while preventing catastrophic forgetting in the lower layers.

Through experimentation we find that unfreezing the top 4 (out of 12) transformer layers provides the best trade-off between model capacity and overfitting risk on this relatively small 9.22-hour dataset.

4.2.2 Temporal Pooling Strategies.

The encoder produces a sequence of hidden states, where a temporal pooling function is needed to produce a fixed-dimensional utterance embedding. We systematically evaluate five pooling strategies:

1. **Mean Pooling**: The masked average over all valid frames (Arora et al., 2017).
2. **Attentive Pooling**: A learnable linear projection computes scalar attention weights over frames, followed by softmax-weighted aggregation.
3. **Learned Attentive Pooling**: An enhanced nonlinear attention mechanism that replaces the single linear projection with a two-layer bottleneck network (Okabe et al., 2018).
4. **Mean + Attentive**: Concatenation of mean-pooled and attentive-pooled vectors.
5. **Learned Mean + Attentive**: Concat of mean-pooled and learned-attentive-pooled vectors.

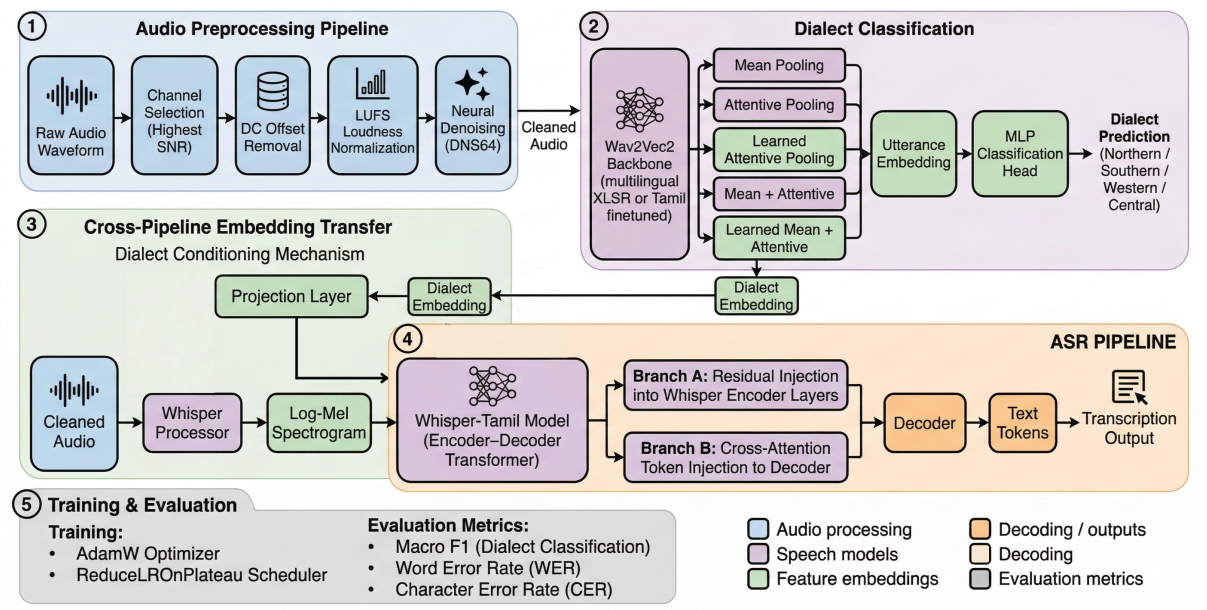


Figure 1: Overview of the proposed dialect-aware Tamil speech processing architecture. Raw audio is first standardized through channel selection, DC offset removal, LUFS normalization, and neural denoising. A Wav2Vec2-based dialect classifier extracts utterance-level dialect embeddings using temporal pooling strategies and an MLP head. These embeddings are transferred across pipelines to condition the Whisper-Tamil ASR model through residual or cross-attention injection, enabling dialect-aware transcription.

4.2.3 Classification Head.

The pooled utterance embedding is passed through a three-layer MLP with progressively decreasing hidden dimensions. The dropout rate of 0.3 is applied after each ReLU activation to regularize the classifier and prevent overfitting.

4.3 Subtask 2: Automatic Speech Recognition

For ASR, we explore both dialect-conditioned and baseline approaches using the Whisper-Tamil models (*vasista22/whisper-tamil-small* and *vasista22/whisper-tamil-medium*), which are fine-tuned variants of OpenAI's Whisper. We compare their performance through experiments to analyze the impact of model scale on dialect-aware ASR.

4.3.1 Dialect Embedding Extraction.

A central component of our dialect-conditioned ASR approach is the extraction of utterance-level dialect embeddings from the trained dialect classifier. Specifically, for each audio sample, we perform a forward pass through the best dialect classifier and extract the output of the learned attentive pooling layer before the MLP classification head. These embeddings capture rich dialect-discriminative information including phonetic quality, prosodic patterns, and speaking rate characteristics of each dialect region in a continuous vector space. The embeddings are pre-computed for all training samples and stored as a NumPy archive for efficient loading during ASR training.

4.3.2 Variant 1: Residual Injection

The pre-extracted dialect embedding is first projected to the Whisper encoder's hidden dimension using a linear layer followed by layer normalization. The projected embedding is then injected into the encoder's residual stream at every transformer layer. Specifically, at each layer ℓ, the dialect embedding is added to the hidden states before the self-attention computation, analogous to the injection of conditional embeddings used in diffusion models. This repeated injection introduces a persistent dialect bias in the encoder representations, guiding the model toward dialect-aware acoustic processing across multiple levels of abstraction.

4.3.3 Variant 2: Cross-Attention Injection.

In this approach, the dialect embedding is projected into a single token and concatenated to the front of the encoder output sequence. The decoder's cross-attention mechanism then attends to this augmented sequence, allowing it to condition the transcription on dialect identity through its attention queries. A staged training schedule is employed: during warmup, the entire Whisper model is frozen and only the dialect projection layer is trained, al-

lowing it to learn a meaningful mapping. After warmup, the decoder and projection output layer are unfrozen for joint fine-tuning.

4.3.4 Variant 3: Vanilla Whisper Baseline.

A standard fine-tuning without any dialect conditioning, serving as a controlled baseline. All model parameters are trainable during this fine-tuning.

5 Experimental Setup

Experiments were conducted on an NVIDIA RTX 4080 Laptop GPU locally and on cloud instances equipped with NVIDIA RTX A5000 GPUs for computationally intensive workloads. Wav2Vec2 normalization was disabled since recordings were already LUFS-normalized. A fixed random seed of 17 was used to ensure reproducibility.

All models were optimized using AdamW (Loshchilov and Hutter, 2019). For dialect classification, differential learning rates were applied: 10^{-5} for unfrozen Wav2Vec2 encoder layers and 10^{-4} for the pooling and MLP layers. For ASR models, a learning rate of 10^{-5} with weight decay 10^{-2} was used. Training employed ReduceLROnPlateau scheduling with a factor of 0.5 and patience of 5 based on validation performance. Dialect classifiers were evaluated on a held-out 20% development split using macro F1 (Van Rijsbergen, 1979) `torchmetrics`), and the best-performing model was used for the final submission. ASR performance was evaluated using Word Error Rate (WER) (Rabiner and Juang, 1993).

6 Results and Analysis

6.1 Subtask 1: Dialect Classification

Our best-performing dialect classification variant used learned attentive pooling with partial fine-tuning of the top four Wav2Vec2 transformer layers. This configuration achieved a macro F1 of **0.79**, placing **1st** with a margin of 0.26 over the next team (Table 2). Learned attentive pooling was particularly effective as it captures temporally localized dialect cues such as formant transitions and intonation rather than averaging representations across all frames.

6.2 Subtask 2: Automatic Speech Recognition

Our submission used the dialect-conditioned Whisper architecture with residual dialect embedding injection and achieved a WER of **0.90**, ranking 8th (Table 3). Compared to the vanilla Whisper baseline, the dialect-conditioned model showed improved robustness to regional pronunciation variations. Although the model demonstrated stable validation behavior during experimentation, it was only partially trained due to time and computational constraints, which likely limited its performance in the shared task. We expect that fully training the dialect-conditioned models—particularly the cross-attention variant—would further reduce WER by guiding the decoder toward dialect-specific vocabulary and acoustic patterns.

Rank	Team	Macro F1
1	**Wise**	**0.79**
2	Wave2Word	0.53
3	IIITK_SpeechScape	0.48

Table 2: Top teams in Subtask 1 (Dialect Classification).

Rank	Team	WER
1	CHMOD_777	0.51
2	CUET_InferX	0.54
3	Wave2Word	0.55
8	**Wise**	**0.90**

Table 3: Top teams in Subtask 2 (Speech Recognition).

7 Discussion

Our core contribution is cross-pipeline dialect embedding transfer: embeddings from the classifier's attentive pooling layer are injected into Whisper via residual injection (at every encoder layer) or cross-attention conditioning (as a decoder context token), with minimal added parameters. The superiority of non-linear learned attentive pooling over mean pooling confirms that dialect cues are sparse and localized, making a trainable frame-scoring network more effective than simple averaging.

8 Conclusion

The **Wise** system achieved **1st place** in Tamil dialect classification (macro F1: 0.79) using learned attentive pooling with partial Wav2Vec2 fine-tuning. We also proposed two dialect-conditioned Whisper architectures (residual injection and cross-attention) for dialect-aware ASR. Future work will complete training of these ASR variants and also address the class imbalance.

Code: `https://github.com/Ganesh2609/DialectBasedSpeechProcessing`

Limitations

1. The training dataset is relatively small (9.22 hours of speech), which limits the ability of large speech models to fully adapt to dialectal variations.
2. The dataset is class-imbalanced (Central: 885 vs. Northern: 1,696 samples), which may reduce recall for underrepresented dialect groups.
3. Some transcriptions in the dataset contain minor inaccuracies or partially incorrect words, which can introduce noise during ASR training and evaluation.

Acknowledgments

We thank the organizers of the DravidianLangTech@ACL 2026 shared task for curating the Tamil dialect speech corpus and providing the evaluation infrastructure.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR*.

Premjith B, Jyothish Lal G, Sowmya V, Bharathi Raja Chakravarthi, Rajeswari Natarajan, Nandhini K, Abirami Murugappan, Bharathi B, Kaushik M, Prasanth Sn, Aswin Raj R, and Vijai Simmon S. 2023. Findings of the shared task on multimodal abusive language detection and sentiment analysis in tamil and malayalam. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460.

B. Bharathi, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, S. Saranya, and S. Suhasini. 2026. Findings in Tamil Dialect Speech Recognition and Classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

B Bharathi, S Saranya, P Vijayalakshmi, and T Nagarajan. 2025. Multi-dialect speech corpus creation for enhancing tamil automatic speech recognition. *Circuits, Systems, and Signal Processing*, pages 1–19.

Alexandre Defossez, Gabriel Synnaeve, and Yossi Adi. 2020. Real time speech enhancement in the waveform domain. *arXiv preprint arXiv:2006.12847*.

Allen Gorin and 1 others. 2014. Investigating multidialectal Arabic acoustic modeling. In *Proceedings of Interspeech*.

Akshit Jain and 1 others. 2020. Contextual semi-supervised learning: An approach to leverage air-surveillance and general data in speech recognition. In *Proceedings of Interspeech*.

R Jairam and 1 others. 2024. CEN_Amrita at DravidianLangTech-2024: Whisper-based dialect identification for Tamil. In *Proceedings of the Fifth Workshop on Speech and Language Technologies for Dravidian Languages*.

Don Johnson. 2006. Signal loudness measurement using the ITU-R BS.1770 standard. *AES Journal*.

Stephen E Levinson. 1986. Continuously variable duration hidden Markov models for automatic speech recognition. *Computer Speech & Language*, 1(1):29–45.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda. 2018. Attentive statistics pooling for deep speaker embedding. In *Interspeech*.

Lawrence R Rabiner and Biing-Hwang Juang. 1993. *Fundamentals of Speech Recognition*. Prentice Hall.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. *Proceedings of the 40th International Conference on Machine Learning*.

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

Ganesh Sundhar S, Durai Singh K, Gnanasabesan G, Hari Krishnan N, and Mc Dhanush. 2025a. Wise@LT-EDI-2025: Combining classical and neural representations with multi-scale ensemble learning for code-mixed hate speech detection. In *Proceedings of the 5th Conference on Language, Data and Knowledge: Fifth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 54–62, Naples, Italy. Unior Press.

Ganesh Sundhar S, Hari Krishnan N, Arun Prasad T D, Shruthikaa V, and Jyothish Lal G. 2025b. CrewX@LT-EDI-2025: Transformer-based Tamil ASR fine-tuning with AVMD denoising and GRU-VAD for enhanced transcription accuracy. In *Proceedings of the 5th Conference on Language, Data and Knowledge: Fifth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 11–16, Naples, Italy. Unior Press.

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*.

C. J. Van Rijsbergen. 1979. *Information Retrieval*. Butterworth-Heinemann.

Felix Weninger, Hakan Erdogan, Shinji Watanabe, and 1 others. 2015. Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR. In *International Conference on Latent Variable Analysis and Signal Separation*.

Author Index

A, Anshid K, 101
A, Moogambigai, 366
A, Naveen, 248
A, Rachana, 306
A, Rameez Mohammed, 316
A, Savith, 279
Abdullah, Taha Ibne, 348
Abhinav, Gulisetty, 232
Abir, Shahad, 222
Adebayo, Kolawole, 6
Adyanthaya, Raksha, 80
Ahmad, Shadab Tanjeed, 442
Alaguraj, Nahul, 258
Aljunid, Mohammed Fadhel, 80
Anilkumar, Arjungopal, 32
Arumugam, Prabalakshmi, 176, 181, 186

B, Bharathi, 71, 148, 227, 354, 366, 392, 397
B, Nandhini, 289
B, Premjith, 32, 101, 114, 124, 376
B, Senthil Kumar, 436
Barua, Mahashweta Manjari, 253
Beinborn, Lisa, 52
Bellamkonda, Sivaiah, 1
Bhuvanesh, Chinthala, 124

C, Anuradha, 414
C, Jerin Mahibha, 279
Chakravarthi, Bharathi Raja, 71, 80, 88, 101, 114, 124, 134
Chandrasekar, Shruti, 301
Chinnan, Shunmuga Priya Muthusamy, 71, 88, 101
Chowdhury, Shiti, 212, 217, 253
CN, Subalalitha, 101

D, Pandiarajan, 366
Datta, Rahul, 348
Debnath, Rotna Dipika, 191
Dey, Abir, 207
Dey, Shuva, 207
Dhamodaran, Harissh Ragav, 268
Dhar, Ratnajit, 201
Durairaj, Thenmozhi, 80, 114, 124, 134
Dutta, Arka, 196

Eid, Tahmima Hoque, 420, 429

Faisal, Adnan, 196
Francis, Meclin A, 144, 263, 326

G, Gnanasabesan, 381, 447
G, Jyothish Lal, 114, 124, 381, 447
G, Moneissh A, 387
G, Vivekananda, 273
Gayathri.k, , 227
Gowda, Anusha M D, 80

Hegde, Parameshwar R, 80
Herygers, Aaricia, 52
Hossain, Md. Ajwad, 341

Islam, Jihadul, 201
Iyer, Nimisha M, 409

J, Janish Andrin, 153
J, Kavinraj, 387
Jana, Abhik, 14
Jananayagan, , 88
Johnson, Joel, 144, 263, 326
Jose, Jobin, 242, 409

K, Durai Singh, 381
K, Jaisanth, 289
K, Kiruthika, 376
K, Sanjay Krishnan, 331, 336
K, Varalakshmi, 148, 392
K.K.Thamizhmathi, , 354
Kabir, Md. Abtahee, 403
Kajanath, Disne, 62
Kamakshi, Vidhya, 21
Karunanidhi, Arunaggiri Pandian, 176, 181, 186
Khanam, Tasnia, 253
KP, Suriya, 381, 447
Krishna, Jayashree, 359
Krubhakaran, Triambiga, 436
Kumar, R Ranjith, 242
Kumar, R Vinay, 158
Kumar, S D Madhu, 316
Kumar, Saurabh, 273
Kumarasamy, Garthigan, 62
Kumaresan, Prasanna Kumar, 80, 114, 134
Kumari, Jyoti, 144, 263, 326

Labiba, Ayesha, 191
Li, Andrew, 169

M, Yazhmozhi V, 42
Madhurapantula, Manohar Sita Rama, 294
Mahmud, Sha Newaz, 207
Majumder, Anindya, 196
Mallik, Arpita, 191, 371
Manimaran, Karthikeyan, 258
Manjula, , 359
Menon, Suryansh Ram, 32
Mohammed, Danish, 21
Mojumder, Priyontee, 371
Moni, Joyeta Barua, 403
Murad, Hasan, 191, 196, 201, 207, 212, 217, 253, 348, 371, 403, 420, 429
Murugan, Shanthi, 414
Muthukumar, Surendhar, 52

N, Ashmi S, 409
N, Balaji, 436
N, Hari Krishnan, 381, 447
Nagarajan, Kaviya, 436
Nali, Bhavana, 14
Nanda, Tanisha, 232
Nandi, Sukumar, 273
Naswan, Ruwad, 442
Nguyen, Dinh Khac Phuc, 321
Nivetha, , 284

Orny, Noore Tamanna, 403

P, Karthiyayini, 248
P, Rathnakara Shetty, 80
P, Vijayalakshmi, 301
Palani, Balasubramanian, 242, 268, 409
Pannerselvam, Kathiravan, 88
Patnaik, Shrey, 279
Ponnusamy, Kishore Kumar, 134
Ponnusamy, Rahul, 101, 124
Prakash, Diya, 242
Pranesh, TT, 354
Pulagara, Seshu Babu, 294

R, Anbuaruvi, 414
R, Hanish Vigneshwar, 258
R, Kaviya Varma, 163
R, Nithya Varshini C N, 331, 336
R, Ramesh Kannan, 88, 232
Rahmat, Nuzha Saifa, 253
Rahul, Chennuru, 6
Rajalakshmi, Ratnavel, 88, 124, 232, 237, 258
Rajamanickam, Siranjeevi, 242, 409
Rajiakodi, Saranya, 88, 101, 114, 124, 134
Rajkumar, Charmathi, 88
Rebayet, Oarisa, 420, 429
Reddy, Bojja Revanth, 1
Rishta, Miftahul Jannat, 212, 217
Robert, Wordson, 279
Ruhi, Shahrin Afroz Hoque, 191

S, Divagar, 32
S, Ganesh Sundhar, 381, 447
S, Inigashree N, 387
S, Kalaivani K, 248, 289
S, Kanimozhi Selvi C, 387
S, Kishore Shankar, 268
S, Nethrasri, 284
S, Nikhil Karthik, 366
S, Praveen Kumar, 242
S, Saranya, 71, 153
S, Shanthi, 163
S, Suhasini, 71
S, Vedajanaani R, 301
S.b.priya, , 397
S.Sumathi, , 331, 336
Sahil, Mohammed, 153
Sai, Chava Shanmukha, 158
Sai, Chava Srinivasa, 158
Sami, Isnat Mehrin, 371
Sanasam, Ranbir Singh, 273
Sankar, Mirudhula, 237
Saravanan, Nandhika, 366
Sarker, Samonwita, 371
Sekar, G Srishtik, 268
Semon, Md. Ashraful Islam, 201
Shashirekha, Hosahalli Lakshmaiah, 306, 359
Sivagnanam, Bhuvaneswari, 88, 101
Sivalingam, Narthana, 62
Sreekumar, Malavika, 144, 263, 326
Subramanian, Malliga, 284
Surapaneni, Jigeesha Sai, 158

T, Roahiyaa, 376
Tabassum, Fawzia, 420, 429
Thangasamy, Sathiyaraj, 134
Tharaniya Sairaj, R, 268
Thayasivam, Uthayasanker, 62
Thevakumar, Jubeerathan, 62

Uddin, Junain, 348
Ulli, Vinay Babu, 144, 263, 326
Uthayakumar, Sathurgini, 62

V, Lavana, 163
V, Mitharshana T, 163
V, Rithikaa, 331, 336
Vegupatti, Mani, 134
Vigneshwaran, S, 354
Visser, Jacky, 42
Văn, Thìn Đặng, 321
Waller, Annalu, 42
Wong, Sidney, 169
Zaman, Sumaiya, 212, 217