SSN-Nova@LT-EDI 2024: Leveraging Vectorisation Techniques in an Ensemble Approach for Stress Identification in Low-Resource Languages

A Ankitha Reddy, Ann Maria Thomas, Pranav Moorthi & B. Bharathi
Department of CSE
Sri Sivasubramaniya Nadar College of Engineering, Tamil Nadu, India
ankithareddy2210178@ssn.edu.in, annthomas2210391@ssn.edu.in
pranav2210176@ssn.edu.in, bharathib@ssn.edu.in

Abstract

This paper presents our submission for Shared task on Stress Identification in Dravidian Languages: StressIdent LT-EDI@EACL2024. The objective of this task is to identify stress levels in individuals based on their social media content. The system is tasked with analyzing posts written in a code-mixed language of Tamil and Telugu and categorizing them into two labels: "stressed" or "not stressed." Our approach aimed to leverage feature extraction and juxtapose the performance of widely used traditional, deep learning and transformer models. Our research highlighted that building a pipeline with traditional classifiers proved to significantly improve their performance, surpassing the baseline as well as deep learning and transformer models.

1 Introduction

Stress is a complex and multifaceted psychological and physiological response to challenges or demands, often characterized by a state of heightened arousal and a perceived inability to cope with the stressor (Yaribeygi et al., 2017). In the contemporary era, psychological stress is widely acknowledged as a major factor contributing to a variety of health issues and mental disorders. The complexities of modern life, marked by rapid technological changes, societal expectations, and economic pressures, have made stress a pervasive issue. People from various backgrounds are experiencing the adverse effects of persistent stress, leading to various health challenges and mental health issues (Lin et al., 2014b).

The dynamic evolution of social networks has prompted a widespread trend wherein individuals extensively employ various social media platforms as primary channels for expressing their thoughts and emotions. This shift in communication patterns emphasizes the growing reliance on these platforms as primary channels for expressing perspectives and feelings. Notably, amidst this surge, people increasingly turn to social media to vent about their stress (Jalonen, 2014), highlighting the importance of identifying and addressing these expressions to support the mental well-being of users. Various approaches have been crafted for the analysis of physiological data with the goal of stress identification (Greene et al., 2016). The feasibility of identifying the stress of the users through the analysis of their social media activities, such as tweets has been substantiated (Lin et al., 2014a). However, the current state of machine learning models reveals a gap in addressing stress detection in Dravidian languages, such as Tamil and Telugu due to the dearth of well-annotated datasets and proficiently trained models specific to these linguistic contexts. This deficit poses a significant challenge in developing accurate algorithms for identifying stress within the nuanced expressions inherent in Dravidian language structures.

Our paper's sequence is as follows - In Section 2, we navigate through existing publications focusing on text classification tasks in low-resource languages. Section 3 undertakes an analysis of the dataset distribution. Our proposed model's methodology is outlined in Section 4. Section 5 examines the performance metrics of our solutions.

2 Related Work

Extensive research in the field of sentiment analysis and text classification has predominantly centred around languages with Latin scripts, such as English and Spanish (Argamon and Koppel, 2013; Muñoz and Iglesias, 2022; Miranda et al., 2023). Not much research has been reported in other languages with some notable ones including Arabic (Aljarah et al., 2021; Al-Hassan and Al-Dossari, 2022) and Dravidian languages (Badjatiya et al., 2017; Banerjee et al., 2020).

To address the major issues regarding Dra-

Proceedings of the Fourth Workshop on Language Technology for Equality, Diversity, Inclusion, pages 216–220

vidian code-mixed and code-switched datasets, transformer-based models like m-BERT, distil-BERT, xlm-RoBERTa, and MuRIL, which are pre-trained on a large corpus of multiple Indian languages, have proven to outperform deep learning (DL) models (Dowlagar and Mamidi, 2021a).

However, aiming to explore and leverage the applications of state-of-the-art technology including transformers and deep-learning in low-resource languages, (Roy et al., 2022) implemented ensemble techniques with the experimental outcomes of the weighted ensemble framework outperforming state-of-the-art models by achieving 0.802 and 0.933 weighted F1-score for Malayalam and Tamil code-mixed datasets. Similar results were produced using a pre-trained multilingual-BERT model with convolution neural networks (Dowlagar and Mamidi, 2021b).

Withal, taking the limited availability of annotated data in Dravidian languages such as Tamil and Telugu (S et al., 2022), traditional machine learning models have outperformed state-of-the-art technology in numerous similar ventures due to their ability to learn linear features from smaller datasets (Saumya et al., 2021; Jauhiainen et al., 2021).

3 Dataset Analysis

The task has been bifurcated based on language into Tamil and Telugu. The provided labels for the data were "Stressed" and "Non stressed". The data distribution is provided in Table 1.

Category	Telugu	Tamil
Non - Stressed	3314	3720
Stressed	1783	1784

Table 1: Data distribution

An examination of the distribution of data offers insight into disparities among classes that may pose potential impediments to the efficacy of models. To address this imbalance and optimize the operational efficiency of our model, we conducted data augmentation on the datasets, which will be elucidated in detail in Section 4

4 Methodology

4.1 Data Augmentation

Data augmentation via back translation was implemented to augment the dataset size. The act of translating a text into another language not only transforms the meaning and semantic value of the sentence, but the subsequent back translation introduces a layer of linguistic diversity. This sophisticated process not only elevates the robustness and generalization of the model but also upholds the context and quality of the text. In light of the inherent imbalances within our dataset, a systematic data augmentation strategy was employed to address label disproportionality. The changes made to the dataset are reflected in Table 2.

Category	Telugu	Tamil
Non - Stressed	3314	3720
Stressed	2283	2284

Table 2: Data distribution after augmentation

4.2 Preprocessing

Data preprocessing is a critical step in optimizing model efficiency and influencing performance metrics. The process involves several key steps: firstly, text normalization, which encompasses expanding contractions and converting text to lowercase, promoting uniform analysis. Following this, removing special characters, symbols, and emojis streamlines the text, reducing the volume for the model to process. Subsequently, the elimination of stop words, those with minimal semantic value, expedites processing and enhances computational efficiency. Lastly, stemming reduces words to their root form, aiding tasks like sentiment analysis by consolidating related words.

4.3 Feature Extraction

1. TF IDF Vectorizer: TF-IDF, or Term Frequency-Inverse Document Frequency, is a technique for creating features from text data by measuring the importance of words in a collection of documents. It assigns higher importance to words exclusive to a small set of documents. The TF-IDF vectorizer matches each feature to a numerical value calculated from its TF-IDF score, obtained by multiplying term frequency and inverse document frequency. This methodology is utilized in this task to convert preprocessed data into structured numerical representations, facilitating the application of natural language processing models to unstructured text data.

2. Word2Vec: Embeddings involve translating categorical variables, like words or phrases, into continuous vectors within a lower-dimensional

space. Widely used in machine learning, these numerical representations adeptly capture semantic nuances and contextual meanings. For instance, word embeddings associate words with vectors in a high-dimensional space, preserving semantic relationships. This conversion is crucial for the compatibility of machine learning algorithms with numerical data, enabling a nuanced understanding of the significance of words within the given context (Chatterjee et al., 2019).

4.4 Multilingual BERT (m-BERT)

m-BERT, falling within the realm of transformer models, has undergone training across 104 languages. As a case-sensitive model, m-BERT is predominantly trained on raw texts with the primary objectives of predicting masked words within sentences and forecasting the subsequent sentence. Leveraging a bidirectional approach for predictions, m-BERT exhibits the capability to discern the semantic nature of sentences across different languages. Through the separation of sentences into two classes and the introduction of a special classification token as the initial token in every sequence, m-BERT achieves classification by adding an embedding to each token.

4.5 CNN-LSTM

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (20, 20000, 300)          6000000

 conv1d (Conv1D)             (20, 20000, 32)           28832

 max_pooling1d (MaxPooling1  (20, 10000, 32)           0
 D)

 dropout (Dropout)           (20, 10000, 32)           0

 lstm (LSTM)                 (20, 300)                 399600

 dropout_1 (Dropout)         (20, 300)                 0

 dense (Dense)               (20, 1)                   301

=================================================================
Total params: 6428733 (24.52 MB)
Trainable params: 428733 (1.64 MB)
Non-trainable params: 6000000 (22.89 MB)
_________________________________________________________________
```

Figure 1: CNN-LSTM Model Architecture

Convolutional Neural Networks (CNNs) are multi-layered artificial neural networks renowned for their ability to discern intricate features from diverse datasets, treating text as one-dimensional signals employing filters akin to image processing. By viewing word sequences as spatial structures, CNNs skillfully identify relationships between different segments of sentences and the semantic similarity between sentences.

Long Short-Term Memory (LSTM) stands as a sophisticated variant of recurrent neural networks (RNNs). Its ability to control the information flow to the cell state, carefully regulated by structures called gates empowers the LSTM to selectively preserve or discard data, thereby enhancing its efficacy in capturing intricate dependencies within sequential data.

The CNN-LSTM model employs convolution for local feature extraction and LSTM for interpreting the text ordering. Tokenisation and embedding precede the model architecture, integrating an Embedding layer, 1D Convolution layer, max pooling and LSTM layer, with a detailed description of the model architecture provided in Figure 1.

4.6 Stacking Classifier

This is a mechanism for amalgamating diverse classification models through the utilization of a final classifier, known as the meta-classifier. The training process involves individual classifiers being trained on the designated dataset, while the meta-classifier is subsequently trained on the predicted class labels. This ensemble learning technique, characterized by its flexibility and adaptability across various machine learning algorithms, designates the individual classifiers as base learners, making it a versatile solution applicable to different problem domains.

On analysis of various traditional ML models, Support Vector Classifier and Random Forest Classifier were incorporated as the base models due to their exceptional performance. Logistic regression was implemented as the meta-classifier in our approach as it establishes a connection between independent variables and a categorical outcome variable by approximating the likelihood that the outcome belongs to a specific class. It helps estimate the outcome variable when presented with new predictive variable values.

5 Results and Analysis

The evaluation of the task is done based on the following performance metrics: macro-average precision, macro-average recall and macro-average F1-score as provided in table 4 and 5. Before this is the comparison of the feature extraction techniques implemented in Table 3.

Model	TF IDF	Word2Vec
Logistic Regression	0.93	0.90
Stacking Classifer	0.95	0.92
CNN LSTM	0.42	0.64
Linear SVC	0.97	0.97

Table 3: Results of models on Telugu validation set with different vectorisation Techniques

On analysis of the precision of each model implemented following the different vectorisation techniques, TF-IDF has emerged as the most efficient. This could plausibly be due to TF-IDF's sparse representation of the document-term matrix, emphasizing the importance of terms based on their frequency and rarity across the corpus. This is beneficial when dealing with low-resource scenarios where datasets are limited, as it helps capture distinctive features. Word2Vec, while powerful, relies on distributed representations and might face challenges in low-resource scenarios where the model struggles to capture diverse semantics due to limited data. This may also affect combination vectorisation techniques as it may introduce complexity and degrade the performance.

However, the outlier in this hypothesis is the superiority of the Word2Vec vectoriser when utilised in the CNN-LSTM model due to its lower-dimensional embeddings and continuous vector representations for words that capture semantic relationships and contextual information, making it more suitable for deep learning models.

Model	Precision	Recall	F1-Score
Logistic Regression	0.91	0.89	0.90
Stacking Classifier	0.98	0.98	0.98
CNN LSTM	0.46	0.68	0.55
Linear SVC	0.92	0.91	0.91

Table 4: Performance of the proposed system using validation data in Tamil code-mixed text

Model	Precision	Recall	F1-Score
Logistic Regression	0.93	0.90	0.91
Stacking Classifer	0.95	0.92	0.93
CNN LSTM	0.42	0.64	0.51
Linear SVC	0.97	0.97	0.97

Table 5: Performance of the proposed system using validation data in Telugu code-mixed text

Deep learning techniques have proven to perform significantly well with longer-length sentences (Yenala et al., 2018). However, most social media comments tend to be much shorter, enabling better performance than traditional models. Hence, we hypothesise that an ensemble of traditional classifiers employing probabilistic and deterministic classifiers would produce better results than deep learning models not only due to the linear relationship of the data but also the usage of shorter sentences.

Parallel to both conventional deep learning methodologies and traditional approaches, the m-BERT transformer model achieved an accuracy of 99.93 on the Tamil validation dataset. Nevertheless, despite its extensive multilingual training, theoretically providing the transformer with a competitive advantage, an investigation revealed that the model exhibited signs of overfitting. This occurred despite deliberate efforts to mitigate overfitting by reducing the model's complexity and implementing data augmentation techniques to address class imbalances in the dataset. This revelation prompted a reevaluation of our methodology, prompting an exploration of alternative strategies to fortify the resilience of our model.

6 Conclusion

In conclusion, our research on stress identification for Dravidian languages has yielded insightful findings across various vectorisation techniques and modelling approaches. The analysis of model accuracy points to TF-IDF as the most efficient vectorisation technique.

The discrepancy in the performance of the m-BERT transformer raises questions about its adaptability in specific linguistic contexts, emphasising the importance of thorough validation and optimisation procedures even in well-established transformer models. Furthermore, the study underscores the nuanced dynamics between the nature of the data and model performance.

In light of the discerned constraints of the m-BERT transformer model, Our investigation reveals that social media comments favour traditional models, leading us to propose an ensemble technique, leveraging both probabilistic and deterministic classifiers to outperform deep learning and stand-alone classifiers. The Voting Classifier emerged as an enticing alternative, not only combatting the challenge of overfitting but also elevating the overall efficacy of our model.

References

Areej Al-Hassan and Hmood Al-Dossari. 2022. Detection of hate speech in arabic tweets using deep learning. *Multimedia systems*, 28(6):1963–1974.

Ibrahim Aljarah, Maria Habib, Neveen Hijazi, Hossam Faris, Raneem Qaddoura, Bassam Hammo, Mohammad Abushariah, and Mohammad Alfawareh. 2021. Intelligent detection of hate speech in arabic social network: A machine learning approach. *Journal of Information Science*, 47(4):483–501.

Shlomo Argamon and Moshe Koppel. 2013. A systemic functional approach to automated authorship analysis. *Journal of Law Policy*, 12:299–315.

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets.

Shubhanker Banerjee, Bharathi Raja Chakravarthi, and John P McCrae. 2020. Comparison of pretrained embeddings to identify hate speech in indian code-mixed text. In *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pages 21–25. IEEE.

Ankush Chatterjee, Umang Gupta, Manoj Kumar Chinnakotla, Radhakrishnan Srikanth, Michel Galley, and Puneet Agrawal. 2019. Understanding emotions in text using deep learning and big data. *Comput. Hum. Behav.*, 93:309–317.

Suman Dowlagar and Radhika Mamidi. 2021a. Edione@ lt-edi-eacl2021: Pre-trained transformers with convolutional neural networks for hope speech detection. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 86–91.

Suman Dowlagar and Radhika Mamidi. 2021b. EDIOne@LT-EDI-EACL2021: Pre-trained transformers with convolutional neural networks for hope speech detection. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 86–91, Kyiv. Association for Computational Linguistics.

Shalom Greene, Himanshu Thapliyal, and Allison Caban-Holt. 2016. A survey of affective computing for stress detection: Evaluating technologies in stress detection for better health. *IEEE Consumer Electronics Magazine*, 5:44–56.

Harri Jalonen. 2014. Social media – an arena for venting negative emotions. *Online Journal of Communication and Media Technologies*, 4:53–70.

Tommi Jauhiainen, Tharindu Ranasinghe, and Marcos Zampieri. 2021. Comparing approaches to dravidian language identification.

Huijie Lin, Jia Jia, Quan Guo, Yuanyuan Xue, Jie Huang, Lianhong Cai, and Ling Feng. 2014a. Psychological stress detection from cross-media microblog data using deep sparse neural network. pages 1–6.

Huijie Lin, Jia Jia, Quan Guo, Yuanyuan Xue, Qi Li, Jie Huang, Lianhong Cai, and Ling Feng. 2014b. User-level psychological stress detection from social media using deep neural network. *MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia*, pages 507–516.

Carlos Henríquez Miranda, German Sanchez-Torres, and Dixon Salcedo. 2023. Exploring the evolution of sentiment in spanish pandemic tweets: A data analysis based on a fine-tuned bert architecture. *Data*, 8(6).

Sergio Muñoz and Carlos A. Iglesias. 2022. A text classification approach to detect psychological stress combining a lexicon-based feature framework with distributional representations. *Information Processing Management*, 59(5):103011.

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

Kayalvizhi S, Thenmozhi Durairaj, Bharathi Raja Chakravarthi, and Jerin Mahibha C. 2022. Findings of the shared task on detecting signs of depression from social media. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 331–338, Dublin, Ireland. Association for Computational Linguistics.

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, Kyiv. Association for Computational Linguistics.

Habib Yaribeygi, Yunes Panahi, Hedayat Sahraei, Thomas P Johnston, and Amirhossein Sahebkar. 2017. The impact of stress on body function: A review. *EXCLI J.*, 16:1057–1072.

Harish Yenala, Ashish Jhanwar, Manoj K. Chinnakotla, and Jay Goyal. 2018. Deep learning for detecting inappropriate content in text. *International Journal of Data Science and Analytics*, 6(4):273–286.