Denoising Word Embeddings Generated from Social Media Text

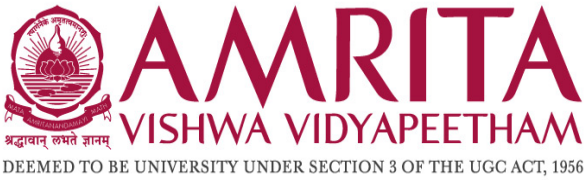
A THESIS

Submitted by

P H V Pavan Kumar
(CB.EN.P2CEN20020)

in partial fulfillment for the award of the degree of

MASTER OF TECHNOLOGY
IN
DATA SCIENCE



Center for Computational Engineering and Networking

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

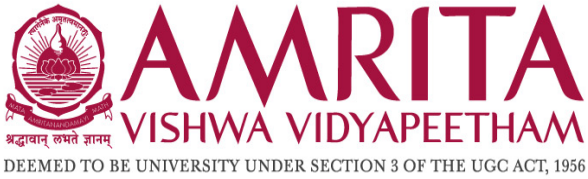
COIMBATORE - 641 112 (INDIA)

JUNE - 2021

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112



BONAFIDE CERTIFICATE

This is to certify that the thesis entitled **"Denoising Word Embeddings Generated from Social Media Text"** submitted by **P H V Pavan Kumar (Register Number-CB.EN.P2CEN20020),** for the award of the **Degree of Master of Technology** in the **"Data Science"** is a bonafide record of the work carried out by him under our guidance and supervision at Amrita School of Engineering, Coimbatore.

Dr. Premjith B
Project Guide

Dr. Sanjanasri.J.P
Project Co-Guide

Dr. K.P.Soman
Professor and Head
CEN

Submitted for the university examination held on 28.06.2022

INTERNAL EXAMINER

EXTERNAL EXAMINER

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112

DECLARATION

I, **P H V PAVAN KUMAR (CB.EN.P2CEN20020),** hereby declare that this thesis entitled **"Denoising Word Embeddings Generated from Social Media Text"**, is the record of the original work done by me under the guidance of **Dr. Premjith B**, Assistant Professor, Centre for Computational Engineering and Networking and **Dr. Sanjanasri.J.P**, Assistant Professor, Centre for Computational Engineering and Networking. To the best of my knowledge this work has not formed the basis for the award of any degree/diploma/ associate ship/fellowship/or a similar award to any candidate in any University.

Date: 28-6-2021 **Signature of the Student**

COUNTERSIGNED

Dr. K.P.Soman

Professor and Head

Center for Computational Engineering and Networking

Contents

Acknowledgement

First of all, I thank Almighty for the immeasurable blessings for smooth completion of my project. I would like to express my deepest gratitude to my guide Dr. Premjith B, Assistant Professor, Computational Engineering and Networking, Amrita Vishwa Vidyapeetham, Coimbatore, for her kindness and constant encouragement and support to complete the project work. It is my genuine pleasure to thank my project co-guide Dr. Sanjanasri.J.P, Assistant Professor, Computational Engineering and Networking, Amrita Vishwa Vidyapeetham, Coimbatore for the continuous motivation and advice for the successful completion of this project.

I thank profusely all the research scholars, Computational Engineering and Networking,Amrita Vishwa Vidyapeetham, Coimbatore, for their kind help throughout my project work. I am grateful to our Head, Dr.K.P.Soman and the Project Co-ordinator, Dr. E.A. Gopalakrishnan and the entire staff of CEN for their timely cooperation

I avail this opportunity to thank my friends for their whole-hearted support during the project. I am greatly indebted to my loving family for their invaluable help, moral support and encouragement throughout the course of study. This accomplishment would not have been possible without them.

List of Figures

List of Abbreviations

NLP	Natural Language Processing
SOTA	state-of-the-art
Mal-Eng	Malayalam-English
Kan-Eng	Kannada-English
Ta-Eng	Tamil-English
MuRIL	Multilingual Representations for Indian Languages
LABSE	Language-agnostic BERT Sentence Embedding
mBERT	Multilingual Bidirectional Encoder Representations from Transformer
S-level	Sentence level
W-level	Word level
SVM	Support Vector Machine
Dtree	Decision Tree
Rforest	Random Forest
KNN	K-Nearest Neighbours

Abstract

The source for noise in word embeddings is numerous. It might be due to the typos in the corpora or during the embedding generation process, including embedding initialisation, mini-batch ordering, etc. Conversations on social media platforms frequently have typos and non-dictionary slang in the text. In this task, we implemented three different Deep learning-based architectures: Deep Neural Network (DNN), Bidirectional-Long Short Term Memory network (Bi-LSTM), and finally, Convolution Neural network (CNN) combined with a Long Short Term Memory network (LSTM) for predicting various sentiments associated with the Dravidian CodeMix languages(Malayalam, Tamil, Kannada). The data given by organizers is highly imbalanced to handle this issue weightage given to each class weight based on their distribution over data. Our experiments reveal that CNN combined with LSTM, DNN with one hidden layer performs best for Malayalam linguistics and, the BiLSTM layer suits the classification of Tamil and Kannada corpus. After inferring the results obtained on performed experiments, we submitted the results. Word embeddings generated from state-of-the-art (SOTA) pre-trained models are also inherent with noise. In this paper, we use Weighted Regularized Least Square (WTR), Fast Fourier Transform (FFT), Wavelet-Based Signal and Variational Mode Decomposition (VMD) algorithms to denoise the word embeddings. This study uses a Sentiment Analysis (SA) CodeMix dataset collected from social media conversations. The dataset comprises of Malayalam-English (Mal-Eng), Kannada-English (Kan-Eng) and Tamil-English (Ta-Eng) languages. Multilingual Representations for Indian Languages (MURIL), Language-agnostic BERT Sentence Embedding (LABSE) and Multilingual Bidirectional Encoder Representations from Transformers (mBERT) are the pre-trained models used for generating embeddings at the Sentence level (S-level) and Word level (W-level). We use Support Vector Machine (SVM), Decision

Tree (Dtree), Random Forest (Rforest) and K-Nearest Neighbours (KNN) classifiers to classify the sentiment of the text. Accuracy, Precision, Recall and F1-score from the classifiers mentioned above are used to measure the improvements in denoising performance. Our experimental results state that the best performance results are obtained for Mal-Eng, at S-level BERT Cased WTR with RF (5% F1-score improvement) and at W-level MuRIL FFT with DTree (2.91% F1-score improvement). For Kan-Eng, mBERT with DTree is the best performing model, and Ta-Eng with FFT denoising shows the best performance with RForest and SVM classifiers.

Chapter 1

Introduction

1.1 Introduction

India is a multilingual country [1] where we often spot conversations on social media platforms [2] like YouTube, Facebook and, Twitter in code-mixed text. *Sentiment analysis* [3] is a concept/technique involved in identifying and analyzing the sentiment/mood of people in the social media [4] context. To classify the underlying sentiments of text as positive, negative, mixed feelings, Native, non-Native, we use sentiment analysis [5].

Text that adopts the vocabulary and grammar from multiple languages frames a new structure based on its usage called code-mixed text [6]. This thesis discusses the methodology and results submitted to the shared task of sentiment analysis for Malayalam-English, Tamil-English, and Kannada-English languages [7]. Three Deep Neural network architectures were implemented for classifying code-mixed text: Convolution Neural Network (CNN) combined with LSTM (CNN-LSTM) [8], Bidirectional-Long Short Term Memory (Bi-LSTM) network [9], and Deep Neural Network(DNN) with one hidden layer.

Word embeddings used in NLP tasks often do not consider the integral noise in data. Avi Caciularu et al. [22] stated that a certain noise level is present in the word embeddings due to the randomized steps used in the word embedding generation process, including embedding initialization and mini-batch ordering negative sampling and subsampling. Similarly, Valentin Malykh et al. [38] considered typos in user-generated sentences as the noise information present in the text. The existence of noise in word embeddings affects the performance of NLP tasks.

Social media is a platform where people from multilingual communities often communicate their opinions by mixing different languages in the same sentence [39, 40]. *Text that adopts the vocabulary and grammar from multiple languages articulates a new structure based on its use called code-mixed text [23, 41].* Conversations on YouTube comments, Twitter tweets, and Facebook posts have non-dictionary slang, typos, and non-standard spelling and grammar variations in their text [36, 39]. However, such issues affect the generation of word embeddings with some inherent noise. In this work, we used the dataset provided by FIRE2021 organizers for the task Sentiment Analysis for Dravidian Languages in Code-Mixed text [23, 33]. The dataset comprises three different languages: Malayalam-English (Mal-Eng), Kannada-English (Kan-Eng) and Tamil-English (Tamil-Eng).

Considering individual word embedding as a one-dimensional (1D) signal, we use traditional approaches WR, FFT and Wavelet Denoising algorithms to denoise the word embeddings. The four pre-trained models:MURIL [10], LABSE [11], BERT Cased and UnCased [12, 42] embeddings at S-level and W-level, are used in this study. Accuracy,

Precision, F1-score and Recall from Support Vector Machine (SVM), Decision Tree (Dtree), Random Forest (RForest) and K-Nearest Neighbour (KNN) are used as evaluation metrics to measure denoising efficacy [43,44]. Based on the study and experiments conducted in this paper, for Mal CodeMix data, embeddings at both the S-level and W-level denoised using the FFT denoising algorithm [45] show the denoising efficacy. Similarly, Kan and Tamil CodeMix embeddings with the WTR denoising algorithm show the denoising efficacy at both the S-level and W-level.

1.2 Literature Survey

In literature, researchers propounded various approaches to remove noise from signals. Some of these techniques comprise least mean square [14, 15], Empirical Mode Decomposition (EMD) [16], Wavelet Transform (mainly comprise Stationary Wavelet Transform (SWT) [17], Discrete Wavelet Transform (DWT) [18], and Fourier Decomposition Method (FDM) [15]). M. Srikanth et al. [19] applied the one-dimensional least square weighted regularized method to two-dimensional image denoising. They achieved significant results in denoising gaussian and salt-and-pepper noised images, surpassing the sparse banded filter-based image denoising. In the same manner, Praveena. R et al. in [15] integrated the 1D least square approach proposed by Selesnick [15] with the concepts of wavelets and achieved impressive results in image denoising.

Yudong Guo et al. [20] developed a denoising autoencoder based on the collaborative filtering algorithm, where they conducted experiments on denoising English corpus. Avi Caciularu et al. considered fusing word embeddings trained on the same corpus with

different initialization for smoothing and improving the quality of word embeddings. Kim Anh Nguyen et al. [22] presented two methods, complete and over complete word denoising embeddings techniques. Both use unsupervised learning to foster the denoised word embeddings generated from SOAT word embeddings. Their analysis of several benchmark tasks shows that the performance of denoised word embeddings surpasses the true word embeddings. Farhad Nooralahzadeh et al. [21] stated that in Named Entity Recognition (NER), false-positive and false-negative instances are the main reason for the reduced performance of auto-generated data. Hence by adopting the approach of partial annotation to address false-negative cases and implementing a reinforcement learning strategy with neural networks, they identify false-positive instances.

1.3 Problem Statement

To minimize the noise in the word embeddings generated from social media conversations.

1. Conversations on social media platforms frequently have typos and non-dictionary slang in the text.

2. The existence of noise entities in data to create word embeddings for the downstream Natural Language Processing (NLP) tasks forms a basis for the presence of noise in the word embeddings.

1.4 Objectives

- To develop deep learning models for sentiment analysis in Dravidian languages.

- To denoise the word embeddings generated from social media conversations using various signal denoising algorithms.
- To study the efficacy of denoised embeddings through extrinsic evaluation by considering sentiment analysis in Dravidian languages as a use case.

Chapter 2

Dataset Description

In this study, we used the dataset provided by the organizers of Dravidian Code-mixed FIRE-2021 [23, 33] for conducting experiments. The data is highly imbalanced with bilingual and native sentiments in Mal-Eng, Kan-Eng and Tamil-Eng languages [34, 35, 37]. To solve the unbalance issue in the dataset, we used stratified sampling to get the balanced data. By classifying each sentence sentiment into five states: unknown_state, Positive, Negative, Mixed feelings, and not-Dravidian in Table 2.1, we detailed the balanced version of the train and test split of data.

Table 2.1: Distribution of train and test dataset over each language.

Sentiments	**Malayalm-English**		**Kanada-English**		**Tamil-English**	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**
unknown_state	900	100	550	50	1600	150
Positive	900	100	550	50	1600	150
Negative	900	100	550	50	1600	150
Mixed_feelings	900	100	550	50	1600	150
not-Dravidian	900	100	550	50	1600	150
Total	**4500**	**500**	**2750**	**250**	**8000**	**750**

Chapter 3

Theoretical Background

3.1 Embedding Models

3.1.1 MURIL

Multilingual Representations for Indian Languages (MuRIL): A Bidirectional Encoder Representations from Transformers (BERT) base architecture model pre-trained on 17 Indian languages and their transliterated counterparts [10]. Google's Indian Research Unit has launched it in the year 2020 and, they have trained the MuRIL model using an existing language learning model called BERT. Its architecture is simple, the model takes the information in the form of tokens and feedforward it to the encoder. Where the encoder converts the tokens to the form of vectors and processes into neural networks, along with this model also needs some metadata at this step.

3.1.2 LaBSE

Language-agnostic BERT Sentence Encoder(LaBSE) is also a BERT-based model trained for sentence embedding for 109 languages [11]. The pre-training process combines masked language modeling with translation language modeling. The model is useful

for getting multilingual sentence embeddings and for bi-text retrieval.

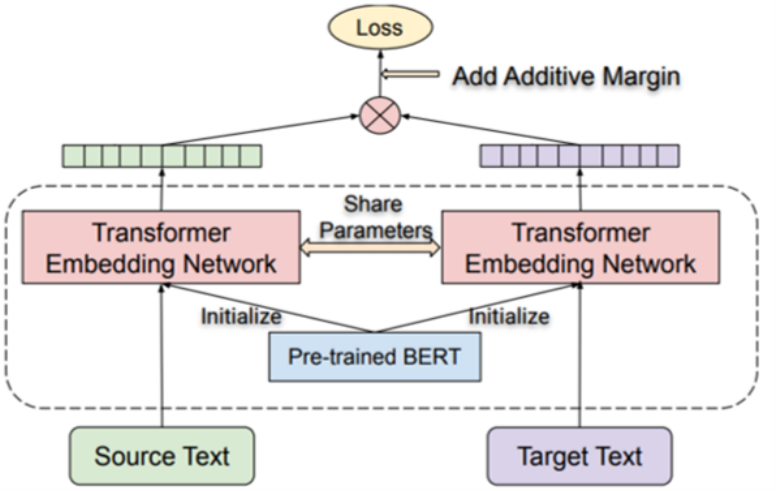


Figure 3.1: Language-agnostic BERT Sentence Encoder architecture.

3.2 Denoising algorithms

3.2.1 Weighted Regularized Least Square Based Denoising

The noise in the 1D signal can be removed through the least square weighted regularization algorithm which was proposed by Ivan. W. Selesnick [14] in his paper.

$$min_x||y - x||_2^2 + \lambda||Dx||_2^2 \tag{3.1}$$

$x \rightarrow$ Denoised signal, $y \rightarrow$ Input Signal $D \rightarrow$ Order of Derivative $\lambda \rightarrow$ Hyper Parameter

In equation 3.1, the L2-norm of the second order derivative of output denoised signal. The L2-norm squared energy of the derivative captures the degree of smoothness. This reduces the level of noise present in the input signal. The minimization of first term in objective function in equation 3.1 forces the output y to be similar to input signal. The

minimization of the second term in equation 3.1 leads to smoothing of the noisy input signal. $\lambda > 0$ is the control parameter used for the least square weighted regularization algorithm. The minimization of the sum in equation 3.1 results in achieving x to be smooth and similar to y.

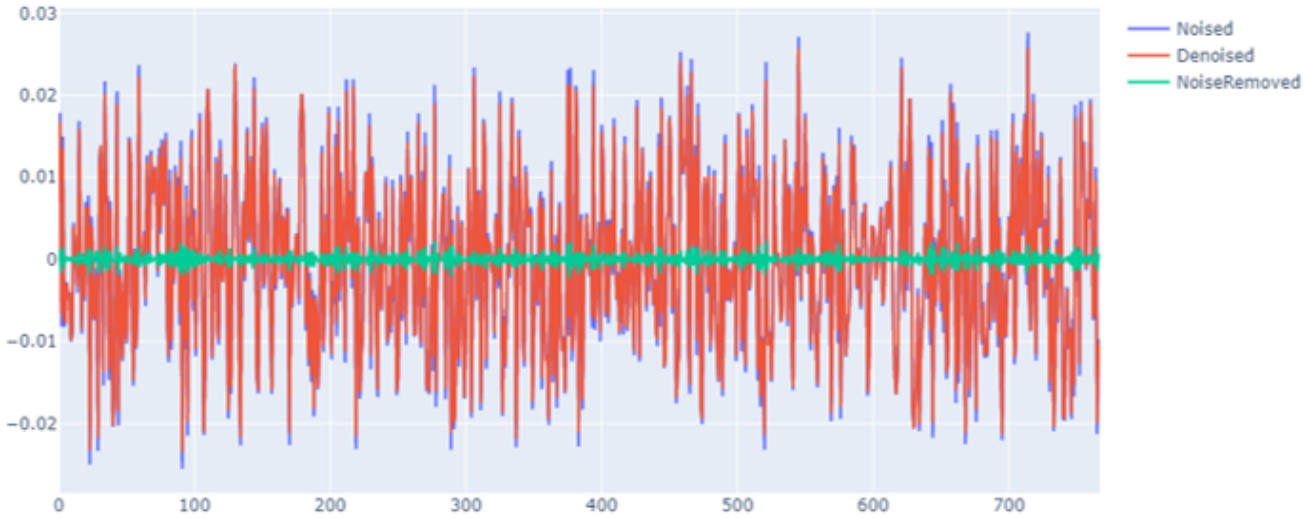


Figure 3.2: WR Denoised MuRIL Word Embedding generated at sentence-level for the Malayalam language.

3.2.2 Fast Fourier Transform Based Denoising

By applying Fast Fourier Transform(FFT)[1] on the embedding generated, we consider only the absolute part of the signal followed by Inverse Fast Fourier Transform(IFFT). The absolute part of the IFFT signal is considered as the denoised signal. Here the FFT analysis is done only in the frequency domain to capture the time series information concept of Wavelets implemented.

[1] https://docs.scipy.org/doc/scipy/reference/tutorial/fft.html

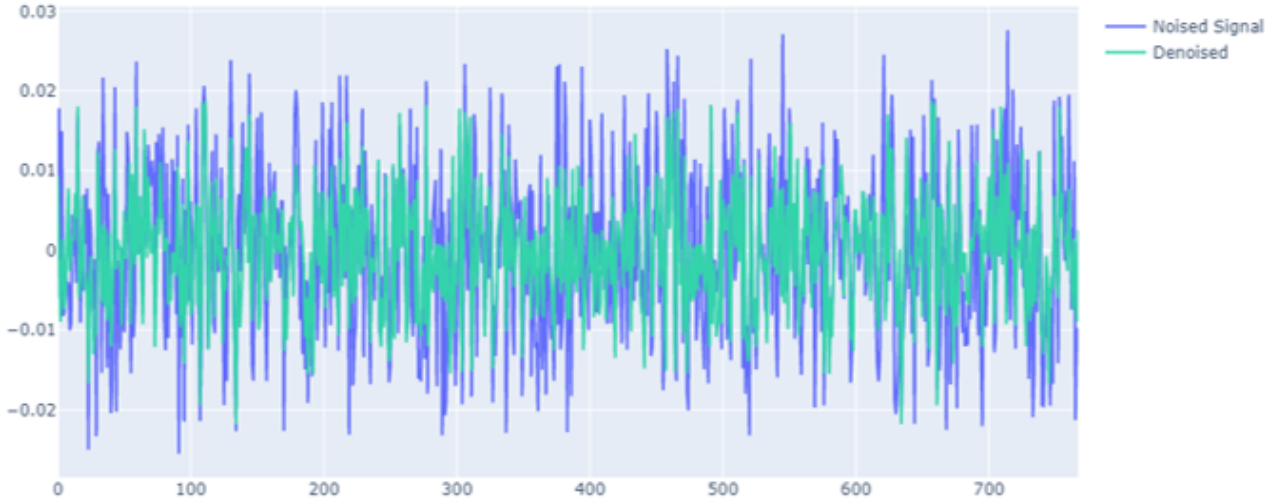


Figure 3.3: FFT Denoised MuRIL Word Embedding generated at sentence-level for the Malayalam language.

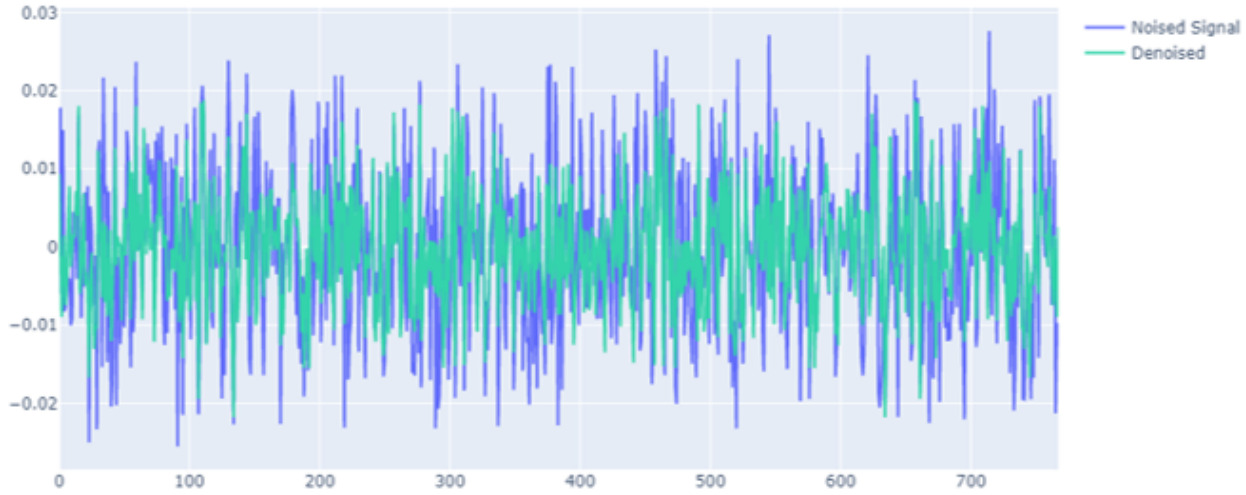


Figure 3.4: Wavelet Denoised MuRIL Word Embedding generated at sentence-level for the Malayalam language.

3.2.3 Wavelet Filtering Based Denoising

As the wavelet can capture the time series information present in the signal, we implemented wavelet functions [25] using the pywt package in python[2]. By applying Discrete Wavelet Transform (DWT) to the embedding signal, we get approximation and detail coefficients only by applying wavelet filters to the detail coefficients we construct the denoised signal with the approximation coefficients.

3.2.4 Variational Mode Decomposition (VMD) Denoising

VMD aims to decompose a real-valued input signal into a number of sub-signals (IMFs) with some sparsity properties of bandwidth in the spectral domain while reproducing the input [26, 28]. It is an adaptive and non-recursive signal decomposition approach in which each mode is almost considered around a centre frequency [27]. The bandwidth of a mode can be assessed by means of constrained variational optimization problem with the following scheme

1. Using Hilbert transform the real signal is transformed into an analytical signal to form a one-sided frequency spectrum of the signal.

2. The frequency spectrum of each mode is shifted by mixing with an exponential function tuned to the respective estimated centre frequency.

3. Using squared norm of the gradient, bandwidth of the mode is estimated.

[2] https://pywavelets.readthedocs.io/en/latest/ref/dwt-discrete-wavelet-transform.html

Considering the problem associated with this work as an unconstrained problem, we define it as below:

$$L\left(\{u_k\},\{w_k\},\lambda\right) = a\sum_k \left\| \partial_t \left[\left(\delta\left(t\right) + \frac{j}{\pi t}\right) * u_k\left(t\right)\right] e^{-jw_k t} \right\|_2^2 + \left\| f\left(t\right) \sum_k u_k\left(t\right) \right\|_2^2 + \left\langle \lambda\left(t\right), f\left(t\right) - \sum_k u_k\left(t\right) - \sum_k u_k\left(t\right) \right\rangle$$

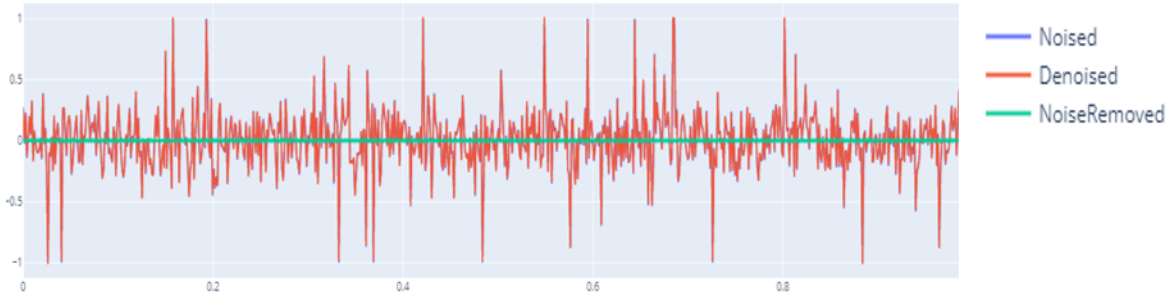


Figure 3.5: VMD Denoised MuRIL Word Embedding generated at sentence-level for the Malayalam language.

$$\hat{u}_k^{n+1}\left(w\right) = \frac{\hat{f}\left(w\right) - \sum_{i=k+1}^{k} \hat{u}_n^k\left(w\right) + \frac{\hat{\lambda}(w)}{2}}{1 + 2\alpha\left(w - w_k^n\right)^2} \tag{3.2}$$

$\alpha \rightarrow$ Balancing parameter of the data-fidelity constrain. $u_k \rightarrow$ ' k^{th} mode of the signal. $\{u_k\} \rightarrow$ set of all modes $\{u_1, u_2.....u_k\}$. $w_k \rightarrow$ center frequencies of k^{th} modes of signal. $\{w_k\} \rightarrow$ set of centre frequencies of the modes. $\delta(t) \rightarrow$ Dirac function. $\hat{f}\left(w\right), \hat{u}\left(w\right), \hat{\lambda}\left(w\right), \hat{u}_k^{n+1}\left(w\right)$ are the Fourier transforms of $f\left(t\right), u\left(t\right), \lambda\left(t\right), u_k^{n+1}\left(t\right)$

Chapter 4

Methodology

This section details on various stages of the methodology followed in denoising word embeddings, as illustrated in Figure 4.1. Initially, raw data is preprocessed and forwarded to the pre-trained models to generate S-level and W-level embeddings. Further, by processing them to the Machine Learning (ML) models [29], Accuracy, Precision, F1-score and Recall are evaluated and baselined to evaluate the efficacy of denoised embeddings.

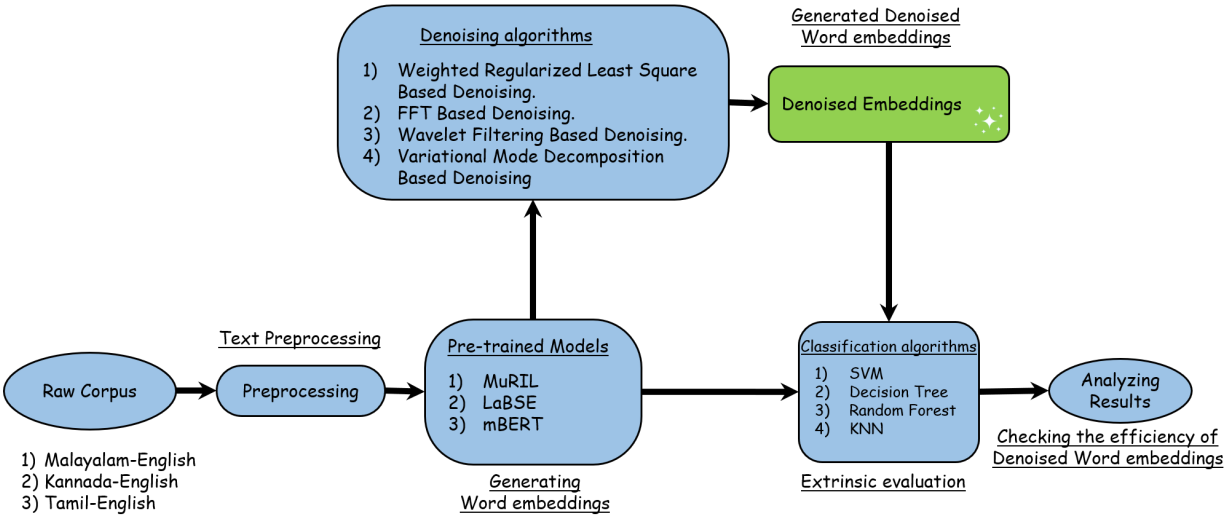


Figure 4.1: Methodology followed in conducting experiments

4.1 Preprocessing

Dataset used in this paper is a part of social media conversations with noise entities (special characters, URLs, and emojis) and stop words associated with the text. The pre-processing [30] stage eliminates noisy entities and the stop words in the corpus. In the next stage, the pre-trained transformer models handle text tokenization and max padding.

4.2 Generating Word embeddings

In this stage, we used four SOTA transformer models pre-trained in Dravidian and English languages to generate sentence and word level embeddings [31, 32].

- For sentence-level embeddings, we input a complete sentence into the model.
- For word-level embeddings, we input each word at the sentence level to the model and the linear combination of word embeddings is done to develop back to a complete sentence.

4.3 Generating Denoised Word embeddings

Considering the output from the word embeddings generation stage as noisy. This stage uses WR, FFT, Wavelet and VMD Denoising techniques to denoise the noisy embeddings. After denoising, we evaluated the classification performance on denoised embeddings in the next stage.

4.4 Classification algorithms

In this stage, we conducted an extrinsic evaluation to review the efficacy of the denoised embeddings generated at the S-level and W-level. We used SVM, Dtree, RForest and KNN classifiers in building the model [29]. As shown in Figure 4.1, the test performance of noised embeddings is initially measured using the built ML models. Following the test performance of denoised embeddings is also measured and compared with the noised embeddings. Accuracy, Precision, recall, and F1-Score are the evaluation metrics in both cases.

Chapter 5

Results and Discussion

This section reports the denoising performance of Mal-Eng, Kan-Eng and Tamil-Eng languages. The experiments conducted and the results inferred in this work are detailed in Table 5.9-5.14.

We used three different deep neural network Models illustrated in Figure **??** to conduct the shared task experiments[1]. Model-1 contains a 1D-CNN, Max Pooling, LSTM layer, and a fully connected dense layer; Model-2 had one Bi-LSTM layer followed by a dense layer; Model-3 had a hidden layer and one fully connected dense layer. The experimental results on the training dataset of all three Models on the selected hyperparameters are in Table 5.2,5.3,5.4, and the validation performance is in Table 5.5. The best-performing Model metrics values are highlighted in bold font.

DNN with one Hidden layer achieve better classification than Model-1 and Model-2 on the Malayalam-English language. BiLSTM with the mentioned hyperparameters in Tabel 5.1 performs better than Model-1 and Model-3 on the Kannada-English CodMix. For the Tamil-English corpus based on training and testing performance and the metric

[1] https://github.com/phvpavankumar/Sentiment-Analysis-for-Malayalam-Tamil-and-Kannada-Languages

Table 5.1: Hyperparameter values and the optimal values used in Model-2&3

	Hyperparameter	Values	Optimal Value
Model-2	Embedding dimension	50, 100	100
	embeddings_initializer	uniform, orthogonal, constant	orthogonal
	embeddings_regularizer	L1, L2	L1
	Number of neurons in LSTM layer	16, 32, 64, 128, 256	32
	Activation Function at hidden layer	Sigmoid, RELU	RELU
	Activation Function at Output layer	Softmax	Softmax
	Optimizer	Adam	Adam
	Loss function	Sparse Categorical Crossentropy, Categorical Crossentropy	Categorical Crossentropy
	learning Rate	0.1, 0.01, 0.001	0.01
	Batch size	16, 32, 64, 80, 128, 132, 256	128
Model-3	Embedding dimension	50, 100	100
	Number of neurons in hidden layer	16, 32, 64, 128, 256	128
	Activation Function at hidden layer	Sigmoid, RELU	RELU
	Activation Function at Output layer	Softmax	Softmax
	Optimizer	Adam	Adam
	Loss function	Sparse Categorical Crossentropy, Categorical Crossentropy	Categorical Crossentropy
	learning Rate	0.1, 0.01, 0.001	0.01
	Batch size	16, 32, 64, 80, 128, 132, 256	64

values, we go for Model-2.

Table 5.2: Training performance on Malayalam-English Dataset for various Models

Model	Accuracy	Precision	Recall	AUC
Model-1	0.925	0.8297	0.7866	0.9633
Model-2	**0.9482**	**0.8881**	**0.848**	**0.9806**
Model-3	0.8428	0.8571	0.2545	0.7657

Table 5.3: Training performance on Tamil-English Dataset for various Models

Model	Accuracy	Precision	Recall	AUC
Model-1	0.9732	0.9473	0.9171	0.9919
Model-2	0.8439	0.7037	0.3778	0.8389
Model-3	**0.9905**	**0.9787**	**0.9737**	**0.9972**

The results of Mal-Eng language at the S-level and W-level are listed in Table 5.9

Table 5.4: Training performance on Kannada-English dataset for various Models

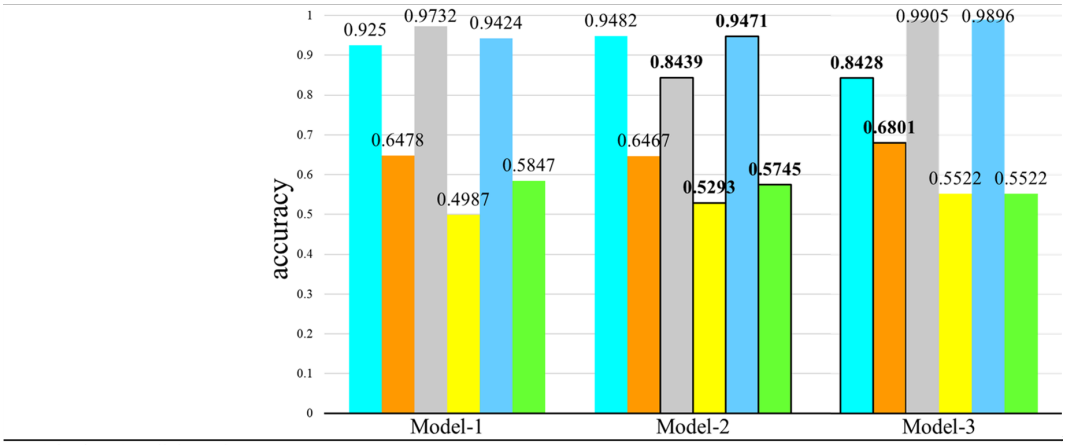
Model	Accuracy	Precision	Recall	AUC
Model-1	0.9424	0.8741	0.8316	0.9769
Model-2	0.9471	0.8823	0.8489	0.9811
Model-3	**0.9896**	**0.9762**	**0.9719**	**0.9992**

Table 5.5: Testing Performance of all the three Models

Language	Malayalam - English			Tamil - English			Kannada - English		
Model	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
Model-1	0.5854	0.6432	0.6077	0.4397	0.5072	0.4384	0.5007	0.5248	0.5085
Model-2	0.5797	0.6346	0.5995	**0.4232**	**0.5072**	**0.441**	**0.5062**	**0.5455**	**0.5193**
Model-3	**0.6303**	**0.6304**	**0.627**	0.43	0.4631	0.4408	0.4855	0.5126	0.4552

and 5.10, respectively. First, we evaluated the classification performance of noised embeddings and compared it with denoised embeddings. We observed that denoised embeddings at S-level and W-level on classification with the Rforest and Dtree show the best improvement in F1-score. Among them, BERT Cased with WTR and MuRIL with FFT show a 5% and 2.91% improvement in F1-score at S-level and W-level, respectively. Finally, as shown in Table 5.8, VMD denoised embeddings with the SVM model give the overall best performance of 46.57% F1-score.

Among all the three datasets Kan-Eng dataset is the smallest one. The results reported in Table 5.11 and 5.12 indicates the classification performance of Kan-Eng denoised embeddings. Initially, WTR denoised embeddings at S-level, we notice MuRIL with Rforest indicates a 4.23% increase in F1-score, but as shown in Table 5.7 with VMD, there is an overall best improvement of 5.18% increase in F1-score. Correspondingly, BERT Cased denoised embeddings with DTree show the best improvement in F1-score at W-level. However, at W-level, LABSE with Wavelet shows no improvement in classification performance before and after denoising.



	Model-1	Model-2	Model-3
Malyalam-English Train accuracy	0.925	0.9482	**0.8428**
Malyalam-English Test accuracy	0.6478	0.6467	**0.6801**
Tamil-English Train accuracy	0.9732	**0.8439**	0.9905
Tamil- EnglishTest accuracy	0.4987	**0.5293**	0.5522
Kannada-English Train accuracy	0.9424	**0.9471**	0.9896
Kannada-English Test accuracy	0.5847	**0.5745**	0.5522

Figure 5.1: Testing Performance of all the three Models

The Tamil-Eng dataset used in this work has more data points than Mal-Eng and Kan-Eng datasets. As mentioned in Table 5.13, FFT denoising on BERT UnCased with Rforest shows a 3.81% improvement in F1-score is the best improvement observed at S-level. Similarly, results mentioned in Tabel 5.14 show that FFT denoising on LABSE with SVM shows the best improvement in F1-score at W-level.

Table 5.6: Malyalam-English VMD denoising results

BERT Cased					
Malayalam		Accuracy	Precision	F1	Recall
Noisy embedding	SVM	46.40%	45.43%	45.00%	46.40%
	DTree	33.20%	33.30%	33.21%	33.20%
	Rforest	43.00%	41.63%	41.90%	43.00%
	KNN	34.40%	36.90%	34.58%	34.40%
WTR Denoising	SVM	46.20%	45.06%	44.58%	46.20%
	DTree	**38.20%**	**38.45%**	**38.26%**	**38.20%**
	Rforest	*45.20%*	*44.05%*	*44.43%*	*45.20%*
	KNN	34.40%	36.57%	34.38%	34.40%
FFT Denoising	SVM	46.60%	46.19%	45.08%	46.60%
	DTree	35.60%	36.46%	35.93%	35.60%
	Rforest	44.00%	42.27%	42.64%	44.00%
	KNN	**36.60%**	**38.62%**	**36.70%**	**36.60%**
Wavelet Denoising	SVM	46.60%	45.69%	45.20%	46.60%
	DTree	34.40%	34.40%	34.36%	34.40%
	Rforest	43.80%	42.24%	42.71%	43.80%
	KNN	34.60%	36.86%	34.64%	34.60%
VMD Denoising	SVM	**47.80%**	**47.54%**	**46.57%**	**47.80%**
	DTree	*36.20%*	*37.27%*	*36.65%*	*36.20%*
	Rforest	**45.40%**	**44.20%**	**44.46%**	**45.40%**
	KNN	*35.60%*	*38.35%*	*35.75%*	*35.60%*

Notes: Accuracy, Precision, F1-score (F1) and Recall of noise and denoised embeddings are reported in the above table. The best and the overall best performance of the model on improvement in F1 for each denoising algorithm are highlighted in italic and Bold, respectively.

Table 5.7: Kannada-English VMD denoising results

MuRIL					
Kannada		Accuracy	Precision	F1	Recall
Bfr Denoising	SVM	48.40%	47.42%	47.38%	48.40%
	DTree	38.40%	39.87%	38.37%	38.40%
	Rforest	48.80%	48.51%	48.25%	48.80%
	KNN	51.20%	52.78%	50.50%	51.20%
WTR Denoising	SVM	*51.60%*	*50.62%*	*50.84%*	*51.60%*
	DTree	*42.00%*	*41.75%*	*39.08%*	*42.00%*
	Rforest	**53.20%**	**52.95%**	**52.48%**	**53.20%**
	KNN	49.20%	50.72%	48.47%	49.20%
FFT Denoising	SVM	**48.80%**	**47.43%**	**47.60%**	**48.80%**
	DTree	*41.20%*	*41.37%*	*39.76%*	*41.20%*
	Rforest	*51.20%*	*51.17%*	*50.52%*	*51.20%*
	KNN	50.00%	51.77%	49.20%	50.00%
Wavelet Denoising	SVM	*46.40%*	*45.21%*	*45.61%*	*46.40%*
	DTree	*42.40%*	*40.77%*	*40.37%*	*42.40%*
	Rforest	*51.20%*	*51.37%*	*50.17%*	*51.20%*
	KNN	45.60%	46.27%	44.45%	45.60%
VMD Denoising	SVM	*43.60%*	*44.72%*	*41.86%*	*43.60%*
	DTree	**43.20%**	**44.08%**	**43.55%**	**43.20%**
	Rforest	*52.80%*	*53.30%*	*52.47%*	*52.80%*
	KNN	**52.00%**	**53.66%**	**51.49%**	**52.00%**

Table 5.8: Tamil-English VMD denoising results

LABSE					
Tamil		Accuracy	Precision	F1	Recall
Noisy embedding	SVM	47.20%	46.38%	46.51%	47.20%
	DTree	33.47%	34.15%	33.61%	33.47%
	Rforest	40.93%	40.56%	40.51%	40.93%
	KNN	40.53%	42.94%	41.17%	40.53%
WTR Denoising	SVM	**48.40%**	**47.80%**	**47.88%**	**48.40%**
	DTree	27.47%	27.54%	27.44%	27.47%
	Rforest	39.20%	38.66%	38.79%	39.20%
	KNN	39.47%	41.83%	40.07%	39.47%
FFT Denoising	SVM	*48.13%*	*47.55%*	*47.68%*	*48.13%*
	DTree	31.20%	31.29%	31.17%	31.20%
	Rforest	40.27%	39.31%	39.51%	40.27%
	KNN	39.60%	41.58%	40.19%	39.60%
Wavelet Denoising	SVM	47.20%	46.38%	46.51%	47.20%
	DTree	28.13%	28.03%	28.05%	28.13%
	Rforest	40.40%	39.85%	39.98%	40.40%
	KNN	40.53%	42.85%	41.14%	40.53%
VMD Denoising	SVM	47.33%	46.56%	46.65%	47.33%
	DTree	**35.20%**	**34.72%**	**34.91%**	**35.20%**
	Rforest	**45.40%**	**43.72%**	**44.17%**	**45.40%**
	KNN	40.53%	42.91%	41.17%	40.53%

Table 5.9: Malyalam-English Sentence level Results

Malayalam		Noisy embedding				WTR Denoising				FFT Denoisng				Wavlet Denoisng			
		Accuracy	Precision	F1	Recall	Accuracy	Precision	F1	Recall	Accuracy	Precision	F1	Recall	Accuracy	Precision	F1	Recall
MuRIL	SVM	33.60%	18.67%	21.47%	33.60%	34.00%	19.59%	22.08%	34.00%	34.20%	19.64%	23.30%	34.20%	*34.40%*	*20.31%*	*23.27%*	*34.40%*
	DTree	38.40%	37.14%	37.28%	38.40%	*42.00%*	*42.33%*	*41.96%*	*42.00%*	*40.00%*	*40.04%*	*39.95%*	*40.00%*	36.80%	37.79%	37.16%	36.80%
	Rforest	49.40%	47.71%	48.11%	49.40%	48.40%	46.90%	46.88%	48.40%	51.00%	49.03%	49.58%	51.00%	49.00%	48.07%	47.43%	49.00%
	KNN	48.60%	48.72%	47.42%	48.60%	47.20%	47.81%	46.14%	47.20%	50.60%	51.38%	49.51%	50.60%	48.40%	48.20%	47.40%	48.40%
LABSE	SVM	57.00%	56.91%	56.85%	57.00%	57.60%	57.52%	57.46%	57.60%	57.60%	57.33%	57.41%	57.60%	57.20%	57.18%	57.08%	57.20%
	DTree	33.20%	34.56%	33.68%	33.20%	*37.80%*	*37.63%*	*37.60%*	*37.80%*	35.40%	36.09%	35.66%	35.40%	37.60%	40.09%	37.70%	37.60%
	Rforest	49.40%	48.70%	48.73%	49.40%	53.00%	52.21%	52.22%	53.00%	*51.80%*	*50.92%*	*51.10%*	*51.80%*	**53.80%**	**53.38%**	**53.36%**	**53.80%**
	KNN	46.80%	51.00%	47.08%	46.80%	47.40%	51.48%	47.78%	47.40%	46.80%	51.31%	47.27%	46.80%	47.00%	51.18%	47.30%	47.00%
BERT Cased	SVM	46.40%	45.43%	45.00%	46.40%	46.20%	45.06%	44.58%	46.20%	46.60%	46.19%	45.08%	46.60%	46.60%	45.69%	45.20%	46.60%
	DTree	33.20%	33.30%	33.21%	33.20%	**38.20%**	**38.45%**	**38.26%**	**38.20%**	*35.60%*	*36.46%*	*35.93%*	*35.60%*	*34.40%*	*34.40%*	*34.36%*	*34.40%*
	Rforest	43.00%	41.63%	41.90%	43.00%	45.20%	44.05%	44.43%	45.20%	44.00%	42.27%	42.64%	44.00%	43.80%	42.24%	42.71%	43.80%
	KNN	34.40%	36.90%	34.58%	34.40%	34.40%	36.57%	34.38%	34.40%	36.60%	38.62%	36.70%	36.60%	34.60%	36.86%	34.64%	34.60%
BERT Uncased	SVM	38.40%	40.75%	33.63%	38.40%	38.40%	40.45%	33.60%	38.40%	37.40%	37.94%	32.48%	37.40%	38.00%	40.62%	33.03%	38.00%
	DTree	29.80%	29.71%	29.69%	29.80%	32.20%	32.17%	32.13%	32.20%	32.00%	31.97%	31.96%	32.00%	*30.20%*	*30.80%*	*30.40%*	*30.20%*
	Rforest	39.00%	37.36%	37.86%	39.00%	*42.00%*	*40.21%*	*40.60%*	*42.00%*	**43.20%**	**41.60%**	**42.00%**	**43.20%**	39.20%	36.58%	37.29%	39.20%
	KNN	33.40%	32.87%	32.30%	33.40%	32.80%	31.97%	31.56%	32.80%	34.40%	33.75%	33.49%	34.40%	32.60%	32.33%	31.66%	32.60%

Notes: Accuracy, Precision, F1-score (F1) and Recall of noise and denoised embeddings are reported in the above table. The best and the overall best performance of the model on improvement in F1 for each denoising algorithm are highlighted in italic and Bold, respectively.

Table 5.10: Malyalam-English World level Results

Malayalam		Noisy embedding				WTR Denoising				FFT Denoisng				Wavlet Denoisng			
		Accuracy	Precision	F1	Recall	Accuracy	Precision	F1	Recall	Accuracy	Precision	F1	Recall	Accuracy	Precision	F1	Recall
MuRIL	SVM	31.40%	43.05%	26.34%	31.40%	31.40%	43.05%	26.34%	31.40%	31.40%	43.04%	26.32%	31.40%	*32.40%*	*43.79%*	*27.13%*	*32.40%*
	DTree	34.00%	34.60%	34.25%	34.00%	35.00%	34.87%	34.93%	35.00%	36.20%	35.67%	35.86%	36.20%	29.20%	29.51%	29.34%	29.20%
	Rforest	37.80%	35.56%	36.25%	37.80%	*39.60%*	*37.64%*	*38.22%*	*39.60%*	**40.40%**	**38.81%**	**39.16%**	**40.40%**	36.40%	34.68%	35.24%	36.40%
	KNN	30.80%	30.22%	30.13%	30.80%	31.20%	30.40%	30.42%	31.20%	32.60%	32.13%	31.81%	32.60%	27.60%	26.84%	26.90%	27.60%
LABSE	SVM	51.20%	50.06%	50.38%	51.20%	51.20%	50.06%	50.39%	51.20%	*51.40%*	*50.19%*	*50.54%*	*51.40%*	51.20%	50.06%	50.38%	51.20%
	DTree	37.80%	37.71%	37.66%	37.80%	37.60%	37.47%	37.49%	37.60%	32.40%	32.34%	32.34%	32.40%	**39.40%**	**38.91%**	**38.70%**	**39.40%**
	Rforest	51.60%	50.17%	50.45%	51.60%	50.80%	49.16%	49.66%	50.80%	49.40%	47.57%	48.11%	49.40%	50.60%	49.44%	49.65%	50.60%
	KNN	51.40%	51.38%	50.54%	51.40%	*52.00%*	*51.83%*	*50.97%*	*52.00%*	49.80%	50.05%	49.21%	49.80%	51.00%	50.83%	50.14%	51.00%
BERT Cased	SVM	39.20%	46.25%	34.48%	39.20%	39.40%	46.49%	34.71%	39.40%	38.40%	39.75%	33.60%	38.40%	39.20%	46.25%	34.48%	39.20%
	DTree	34.80%	34.35%	34.54%	34.80%	34.60%	33.91%	34.20%	34.60%	31.00%	32.12%	31.43%	31.00%	32.20%	32.34%	32.26%	32.20%
	Rforest	45.00%	42.53%	43.10%	45.00%	*45.00%*	*42.88%*	*43.49%*	*45.00%*	*47.60%*	*45.18%*	*45.52%*	*47.60%*	43.80%	41.20%	41.78%	43.80%
	KNN	37.80%	38.06%	37.14%	37.80%	37.80%	38.21%	37.17%	37.80%	38.20%	38.81%	37.61%	38.20%	37.80%	38.06%	37.14%	37.80%
BERT Uncased	SVM	31.80%	44.70%	26.63%	31.80%	31.80%	44.70%	26.63%	31.80%	31.20%	43.99%	26.01%	31.20%	31.80%	44.70%	26.63%	31.80%
	DTree	32.80%	32.29%	32.50%	32.80%	32.20%	32.23%	32.19%	32.20%	*35.00%*	*34.48%*	*34.70%*	*35.00%*	32.60%	32.30%	32.41%	32.60%
	Rforest	42.80%	40.61%	41.27%	42.80%	**45.20%**	**42.83%**	**43.50%**	**45.20%**	43.20%	41.41%	41.89%	43.20%	*43.60%*	*41.45%*	*42.12%*	*43.60%*
	KNN	36.40%	36.25%	36.01%	36.40%	37.00%	36.71%	36.56%	37.00%	37.20%	36.97%	36.91%	37.20%	36.60%	36.53%	36.25%	36.60%

Table 5.11: Kannada-English Sentence level Results

Kannada		Noisy embedding				WTR Denoising				FFT Denoisng				Wavlet Denoisng			
		Accuracy	**Precision**	**F1**	**Recall**	**Accuracy**	**Precision**	**F1**	**Recall**	**Accuracy**	**Precision**	**F1**	**Recall**	**Accuracy**	**Precision**	**F1**	**Recall**
MuRIL	**SVM**	48.40%	47.42%	47.38%	48.40%	51.60%	50.62%	50.84%	51.60%	48.80%	47.43%	47.60%	48.80%	46.40%	45.21%	45.61%	46.40%
	DTree	38.40%	39.87%	38.37%	38.40%	42.00%	41.75%	39.08%	42.00%	41.20%	41.37%	39.76%	41.20%	*42.40%*	*40.77%*	*40.37%*	*42.40%*
	Rforest	48.80%	48.51%	48.25%	48.80%	*53.20%*	*52.95%*	*52.48%*	*53.20%*	*51.20%*	*51.17%*	*50.52%*	*51.20%*	51.20%	51.37%	50.17%	51.20%
	KNN	51.20%	52.78%	50.50%	51.20%	49.20%	50.72%	48.47%	49.20%	50.00%	51.77%	49.20%	50.00%	45.60%	46.27%	44.45%	45.60%
LABSE	**SVM**	58.00%	58.54%	57.88%	58.00%	57.60%	58.10%	57.47%	57.60%	53.60%	53.55%	53.31%	53.60%	56.40%	56.92%	56.25%	56.40%
	DTree	41.20%	42.89%	41.53%	41.20%	41.20%	41.02%	40.85%	41.20%	42.80%	46.13%	42.28%	42.80%	35.60%	36.44%	35.74%	35.60%
	Rforest	52.00%	52.18%	51.44%	52.00%	*54.40%*	*54.60%*	*54.09%*	*54.40%*	**55.60%**	**56.39%**	**55.27%**	**55.60%**	46.40%	47.01%	46.14%	46.40%
	KNN	47.20%	49.42%	46.65%	47.20%	46.40%	49.40%	45.58%	46.40%	46.00%	47.92%	45.24%	46.00%	46.80%	50.54%	45.73%	46.80%
BERT Cased	**SVM**	47.60%	46.67%	45.85%	47.60%	46.40%	45.26%	44.77%	46.40%	46.40%	45.40%	44.75%	46.40%	49.60%	48.91%	47.82%	49.60%
	DTree	31.20%	31.15%	31.13%	31.20%	*32.80%*	*32.90%*	*32.83%*	*32.80%*	33.60%	33.31%	33.03%	33.60%	30.00%	30.31%	30.07%	30.00%
	Rforest	44.00%	43.58%	42.90%	44.00%	42.80%	42.12%	41.56%	42.80%	*46.00%*	*45.65%*	*45.03%*	*46.00%*	**46.80%**	**46.81%**	**46.41%**	**46.80%**
	KNN	40.80%	44.14%	39.48%	40.80%	40.40%	40.77%	38.96%	40.40%	39.20%	43.26%	37.82%	39.20%	41.20%	43.55%	40.19%	41.20%
BERT Uncased	**SVM**	39.20%	32.29%	34.66%	39.20%	39.20%	32.47%	34.74%	39.20%	39.20%	32.28%	34.63%	39.20%	38.00%	34.01%	34.13%	38.00%
	DTree	30.40%	30.33%	30.22%	30.40%	**35.60%**	**35.00%**	**35.21%**	**35.60%**	27.60%	28.19%	27.54%	27.60%	*31.60%*	*32.07%*	*31.68%*	*31.60%*
	Rforest	43.20%	42.59%	42.65%	43.20%	45.20%	45.32%	45.02%	45.20%	*46.40%*	*45.74%*	*45.67%*	*46.40%*	43.60%	44.33%	43.46%	43.60%
	KNN	38.00%	38.77%	37.53%	38.00%	36.80%	37.89%	36.53%	36.80%	38.80%	39.22%	38.26%	38.80%	36.40%	34.11%	34.16%	36.40%

Table 5.12: Kannada-English Word level Results

Kannada		Noisy embedding				WTR Denoising				FFT Denoisng				Wavlet Denoisng			
		Accuracy	**Precision**	**F1**	**Recall**	**Accuracy**	**Precision**	**F1**	**Recall**	**Accuracy**	**Precision**	**F1**	**Recall**	**Accuracy**	**Precision**	**F1**	**Recall**
MuRIL	**SVM**	41.60%	44.38%	39.24%	41.60%	42.00%	44.93%	39.76%	42.00%	43.60%	45.88%	41.51%	43.60%	41.20%	41.34%	39.80%	41.20%
	DTree	41.60%	41.26%	36.30%	41.60%	*40.80%*	*41.25%*	*38.14%*	*40.80%*	*40.80%*	*45.09%*	*39.76%*	*40.80%*	41.20%	39.02%	35.76%	41.20%
	Rforest	44.80%	44.09%	44.15%	44.80%	45.60%	44.60%	44.62%	45.60%	46.80%	45.63%	45.61%	46.80%	*47.20%*	*46.09%*	*45.93%*	*47.20%*
	KNN	44.40%	45.62%	44.44%	44.40%	44.00%	45.35%	44.10%	44.00%	44.00%	44.60%	43.91%	44.00%	42.80%	43.67%	43.03%	42.80%
LABSE	**SVM**	48.00%	49.97%	47.16%	48.00%	47.60%	49.72%	46.82%	47.60%	46.40%	48.72%	45.58%	46.40%	48.00%	49.97%	47.16%	48.00%
	DTree	39.20%	39.47%	39.21%	39.20%	36.00%	36.65%	36.19%	36.00%	36.00%	36.47%	36.17%	36.00%	34.80%	34.55%	34.52%	34.80%
	Rforest	51.20%	51.36%	50.96%	51.20%	*52.40%*	*53.27%*	*52.17%*	*52.40%*	*51.60%*	*51.39%*	*51.10%*	*51.60%*	50.00%	51.01%	49.72%	50.00%
	KNN	48.00%	48.45%	47.28%	48.00%	47.20%	47.68%	46.57%	47.20%	45.60%	45.51%	45.10%	45.60%	46.40%	46.52%	45.77%	46.40%
BERT Cased	**SVM**	49.60%	49.38%	49.16%	49.60%	48.40%	48.03%	47.78%	48.40%	37.60%	37.07%	34.45%	37.60%	34.80%	33.41%	31.40%	34.80%
	DTree	35.20%	36.67%	35.71%	35.20%	**40.00%**	**39.82%**	**39.86%**	**40.00%**	**40.00%**	**40.16%**	**39.86%**	**40.00%**	**40.80%**	**44.19%**	**39.00%**	**40.80%**
	Rforest	46.00%	45.99%	45.63%	46.00%	44.40%	43.86%	43.49%	44.40%	46.80%	46.26%	46.27%	46.80%	49.20%	49.16%	48.08%	49.20%
	KNN	37.20%	39.79%	37.90%	37.20%	36.00%	38.52%	36.67%	36.00%	37.60%	40.41%	38.04%	37.60%	36.00%	38.42%	36.47%	36.00%
BERT Uncased	**SVM**	50.40%	51.56%	50.32%	50.40%	*50.80%*	*51.94%*	*50.85%*	*50.80%*	*52.80%*	*53.15%*	*52.74%*	*52.80%*	52.80%	52.37%	51.72%	52.80%
	DTree	42.00%	42.52%	42.02%	42.00%	42.80%	42.87%	41.10%	42.80%	42.40%	41.93%	40.64%	42.40%	42.00%	45.31%	41.27%	42.00%
	Rforest	48.00%	47.95%	47.62%	48.00%	48.80%	47.99%	47.98%	48.80%	47.60%	48.02%	47.22%	47.60%	46.40%	46.36%	45.78%	46.40%
	KNN	37.20%	38.53%	37.38%	37.20%	36.80%	38.30%	37.07%	36.80%	36.80%	37.54%	36.90%	36.80%	*40.40%*	*41.13%*	*40.10%*	*40.40%*

Table 5.13: Tamil-English Sentence level Results

Tamil		Noisy embedding				WTR Denoising				FFT Denoisng				Wavlet Denoisng			
		Accuracy	Precision	F1	Recall	Accuracy	Precision	F1	Recall	Accuracy	Precision	F1	Recall	Accuracy	Precision	F1	Recall
MuRIL	SVM	46.00%	45.58%	45.68%	46.00%	45.46%	45.07%	45.13%	45.47%	44.80%	44.03%	44.31%	44.80%	44.80%	44.22%	44.37%	44.80%
	DTree	30.13%	30.05%	30.08%	30.13%	*31.20%*	*31.21%*	*31.19%*	*31.20%*	27.20%	27.35%	27.25%	27.20%	*32.00%*	*32.30%*	*31.55%*	*32.00%*
	Rforest	42.53%	41.52%	41.76%	42.53%	42.40%	41.41%	41.52%	42.40%	40.53%	39.45%	39.67%	40.53%	40.80%	39.57%	39.20%	40.80%
	KNN	40.93%	41.46%	40.84%	40.93%	40.40%	41.09%	40.33%	40.40%	39.87%	40.40%	39.95%	39.87%	38.13%	38.48%	37.91%	38.13%
LABSE	SVM	47.20%	46.38%	46.51%	47.20%	*48.40%*	*47.80%*	*47.88%*	*48.40%*	*48.13%*	*47.55%*	*47.68%*	*48.13%*	47.20%	46.39%	46.51%	47.20%
	DTree	33.47%	34.15%	33.61%	33.47%	27.47%	27.54%	27.44%	27.47%	31.20%	31.29%	31.17%	31.20%	28.13%	28.03%	28.05%	28.13%
	Rforest	40.93%	40.56%	40.51%	40.93%	39.20%	38.66%	38.79%	39.20%	40.27%	39.31%	39.51%	40.27%	40.40%	39.85%	39.98%	40.40%
	KNN	40.53%	42.94%	41.17%	40.53%	39.47%	41.83%	40.07%	39.47%	39.60%	41.58%	40.19%	39.60%	40.53%	42.85%	41.14%	40.53%
BERT Cased	SVM	40.93%	40.15%	39.40%	40.93%	41.20%	40.33%	39.67%	41.20%	*42.27%*	*41.72%*	*41.66%*	*42.27%*	41.07%	40.31%	39.63%	41.07%
	DTree	26.40%	26.64%	26.48%	26.40%	**28.00%**	**28.28%**	**28.09%**	**28.00%**	27.33%	27.53%	27.31%	27.33%	**28.27%**	**28.02%**	**28.12%**	**28.27%**
	Rforest	38.00%	36.81%	36.59%	38.00%	38.53%	37.05%	37.12%	38.53%	36.40%	35.65%	35.35%	36.40%	38.00%	36.99%	37.04%	38.00%
	KNN	31.47%	33.17%	31.57%	31.47%	31.87%	34.02%	32.08%	31.87%	31.47%	33.97%	31.73%	31.47%	31.87%	33.81%	32.03%	31.87%
BERT Uncased	SVM	39.20%	38.96%	38.99%	39.20%	39.20%	38.96%	38.99%	39.20%	37.60%	36.03%	36.04%	37.60%	33.20%	30.36%	29.16%	33.20%
	DTree	29.47%	29.04%	28.77%	29.47%	29.47%	29.04%	28.77%	29.47%	28.00%	27.48%	27.56%	28.00%	27.33%	27.50%	27.40%	27.33%
	Rforest	35.87%	34.43%	34.39%	35.87%	*36.80%*	*35.66%*	*35.35%*	*36.80%*	**38.80%**	**38.25%**	**38.20%**	**38.80%**	35.07%	34.30%	34.51%	35.07%
	KNN	32.40%	33.52%	32.56%	32.40%	32.40%	33.30%	32.47%	32.40%	31.73%	33.09%	31.85%	31.73%	32.27%	33.31%	32.40%	32.27%

Table 5.14: Tamil-English Word level Results

Tamil		Noisy embedding				WTR Denoising				FFT Denoisng				Wavlet Denoisng			
		Accuracy	Precision	F1	Recall	Accuracy	Precision	F1	Recall	Accuracy	Precision	F1	Recall	Accuracy	Precision	F1	Recall
MuRIL	SVM	25.60%	21.60%	21.99%	25.60%	25.73%	21.71%	22.11%	25.73%	25.60%	21.60%	21.99%	25.60%	*25.87%*	*21.98%*	*22.44%*	*25.87%*
	DTree	29.47%	30.04%	29.69%	29.47%	29.33%	28.73%	28.97%	29.33%	26.67%	27.12%	26.85%	26.67%	25.20%	25.44%	25.28%	25.20%
	Rforest	31.73%	30.41%	30.74%	31.73%	*32.53%*	*30.98%*	*31.46%*	*32.53%*	*32.67%*	*31.78%*	*31.94%*	*32.67%*	31.60%	30.75%	30.99%	31.60%
	KNN	24.80%	24.73%	24.42%	24.80%	24.53%	24.41%	24.13%	24.53%	23.60%	23.51%	23.14%	23.60%	24.67%	24.94%	24.29%	24.67%
LABSE	SVM	45.73%	44.63%	44.85%	45.73%	46.00%	44.99%	45.18%	46.00%	**47.07%**	**46.37%**	**46.51%**	**47.07%**	45.07%	44.06%	44.26%	45.07%
	DTree	34.67%	36.64%	34.40%	34.67%	*35.33%*	*35.84%*	*35.47%*	*35.33%*	34.53%	39.28%	33.72%	34.53%	34.67%	36.64%	34.40%	34.67%
	Rforest	44.00%	43.41%	43.52%	44.00%	43.60%	42.87%	43.01%	43.60%	44.80%	44.29%	44.47%	44.80%	42.93%	42.24%	42.18%	42.93%
	KNN	40.67%	41.60%	40.95%	40.67%	40.00%	40.92%	40.28%	40.00%	39.73%	41.02%	40.00%	39.73%	40.67%	41.60%	40.95%	40.67%
BERT Cased	SVM	35.33%	40.61%	32.46%	35.33%	35.33%	40.56%	32.43%	35.33%	35.33%	42.15%	32.40%	35.33%	35.33%	40.61%	32.46%	35.33%
	DTree	31.73%	32.21%	31.90%	31.73%	27.20%	27.58%	27.34%	27.20%	27.07%	27.39%	27.16%	27.07%	*32.80%*	*32.78%*	*32.74%*	*32.80%*
	Rforest	39.33%	38.59%	38.70%	39.33%	39.33%	38.44%	38.63%	39.33%	*40.40%*	*39.46%*	*39.55%*	*40.40%*	39.47%	38.77%	38.93%	39.47%
	KNN	33.87%	36.51%	34.25%	33.87%	*34.40%*	*36.83%*	*34.73%*	*34.40%*	32.13%	34.36%	32.15%	32.13%	33.87%	36.51%	34.25%	33.87%
BERT Uncased	SVM	30.80%	25.18%	26.99%	30.80%	30.67%	25.10%	26.89%	30.67%	30.13%	24.82%	26.51%	30.13%	30.80%	25.18%	26.99%	30.80%
	DTree	27.20%	27.18%	27.16%	27.20%	**29.33%**	**29.45%**	**29.37%**	**29.33%**	27.20%	27.29%	27.24%	27.20%	24.67%	24.85%	24.68%	24.67%
	Rforest	35.20%	34.09%	34.22%	35.20%	35.07%	33.71%	33.99%	35.07%	35.60%	33.99%	34.30%	35.60%	**36.13%**	**35.00%**	**35.08%**	**36.13%**
	KNN	29.47%	31.06%	29.80%	29.47%	29.87%	31.76%	30.23%	29.87%	*30.67%*	*31.50%*	*30.65%*	*30.67%*	29.33%	30.92%	29.67%	29.33%

Chapter 6

Conclusion

In this work, we conducted an extrinsic comparative study on denoised embeddings performance at the S- and W-level. This study uses three CodeMix datasets in Mal-Eng, Kan-Eng and Tamil-Eng languages. Initially, we considered that S-level and W-level embeddings from MuRIL, LABSE, BERT Cased, and UnCased are inherent in noise. Considering traditional WR, FFT and Wavelet denoising algorithms, we denoised the embeddings. This study used SVM, Dtree, RForest and KNN as ML classifiers. Baselining the Accuracy, Precession, F1-Score and Recall of noised embeddings, we evaluated the performance of denoised embeddings.

Our experiments state that the classification performance of denoised embeddings over the noised embeddings was noticeable in a few cases. The Mal-Eng FFT denoised embeddings with Rforest and DTree show a 5% and 2.91% improvement in F1-score at the S-level and W-level. In the case of Kan-Eng at the S-level, BERT Uncased WTR embeddings show a 5% F1-score improvement. At W-level, BERT UnCased denoised embeddings show the best denoising performance. Finally, Tamil-Eng language embeddings from BERT UnCased, and LABSE with FFT shows an improvement of 3.81%

and 1.66% at S-level and W-level, respectively. Furthermore, we are interested in denoising the embeddings with advanced denoising techniques ex: By using auto-encoder models.

References

1. Krishnamurti, Bhadriraju. The dravidian languages. Cambridge University Press, 2003.

2. Suryawanshi, Shardul, and Bharathi Raja Chakravarthi. "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, pp. 126-132. 2021.

3. Choudhary, Nurendra, Rajat Singh, Ishita Bindlish, and Manish Shrivastava. "Sentiment analysis of code-mixed languages leveraging resource rich languages." arXiv preprint arXiv:1804.00806 (2018).

4. Banerjee, Shubhanker, Bharathi Raja Chakravarthi, and John P. McCrae. "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), pp. 21-25. IEEE, 2020.

5. Chakravarthi, Bharathi Raja, Ruba Priyadharshini, Vigneshwaran Muralidaran, Shardul Suryawanshi, Navya Jose, Elizabeth Sherly, and John P. McCrae. "Overview

of the track on sentiment analysis for dravidian languages in code-mixed text." In Forum for information retrieval evaluation, pp. 21-24. 2020.

6. Kumaresan, Prasanna Kumar, Ratnasingam Sakuntharaj, Sajeetha Thavareesan, Subalalitha Navaneethakrishnan, Anand Kumar Madasamy, Bharathi Raja Chakravarthi, and John P. McCrae. "Findings of shared task on offensive language identification in Tamil and Malayalam." In Forum for Information Retrieval Evaluation, pp. 16-18. 2021.

7. Banerjee, Shubhanker, Arun Jayapal, and Sajeetha Thavareesan. "NUIG-Shubhanker@ Dravidian-CodeMix-FIRE2020: sentiment analysis of code-mixed dravidian text using XLNet." arXiv preprint arXiv:2010.07773 (2020).

8. Sreelakshmi, K., B. Premjith, and Soman Kp. "Amrita_CEN_NLP@ DravidianLangTech-EACL2021: deep learning-based offensive language identification in Malayalam, Tamil and Kannada." In Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pp. 249-254. 2021.

9. Premjith, B., and K. P. Soman. "Deep learning approach for the morphological synthesis in malayalam and tamil at the character level." Transactions on Asian and Low-Resource Language Information Processing 20, no. 6 (2021): 1-17.

10. Khanuja, Simran, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam et al. "Muril: Multilingual representations for indian languages." arXiv preprint arXiv:2103.10730 (2021).

11. Feng, Fangxiaoyu, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. "Language-agnostic bert sentence embedding." arXiv preprint arXiv:2007.01852 (2020).

12. Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

13. Naveed, Khuram, Muhammad Tahir Akhtar, Muhammad Faisal Siddiqui, and Naveed ur Rehman. "A statistical approach to signal denoising based on data-driven multiscale representation." Digital Signal Processing 108 (2021): 102896.

14. Selesnick, Ivan. "Least squares with examples in signal processing." Connexions 4 (2013).

15. Praveena, R., V. Sowmya, and K. P. Soman. "Least Square Based Signal Denoising Using Wavelet Filters."

16. Naveed, Khuram, Muhammad Tahir Akhtar, Muhammad Faisal Siddiqui, and Naveed ur Rehman. "A statistical approach to signal denoising based on data-driven multiscale representation." Digital Signal Processing 108 (2021): 102896.

17. Kumar, Ashish, Harshit Tomar, Virender Kumar Mehla, Rama Komaragiri, and Manjeet Kumar. "Stationary wavelet transform based ECG signal denoising method." ISA transactions 114 (2021): 251-262.

18. Alfaouri, Mikhled, and Khaled Daqrouq. "ECG signal denoising by wavelet transform thresholding." American Journal of applied sciences 5, no. 3 (2008): 276-281.

19. Azad, Murari Lal, Soumya Das, Pradip Kumar Sadhu, Biplab Satpati, Anagh Gupta, and P. Arvind. "P&O algorithm based MPPT technique for solar PV system under different weather conditions." In 2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT), pp. 1-5. IEEE, 2017.

20. Xu, Bing. "2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC 2017)."

21. Nooralahzadeh, Farhad, Jan Tore Lønning, and Lilja Øvrelid. "Reinforcement-based denoising of distantly supervised NER with partial annotation." Association for Computational Linguistics, 2019.

22. Caciularu, Avi, Ido Dagan, and Jacob Goldberger. "Denoising word embeddings by averaging in a shared space." arXiv preprint arXiv:2106.02954 (2021).

23. Priyadharshini, Ruba, Bharathi Raja Chakravarthi, Sajeetha Thavareesan, Dhivya Chinnappa, Durairaj Thenmozhi, and Rahul Ponnusamy. "Overview of the DravidianCodeMix 2021 shared task on sentiment detection in Tamil, Malayalam, and Kannada." In Forum for Information Retrieval Evaluation, pp. 4-6. 2021.

24. Oliver, Francis Wall, and Arthur G. Tansley. "Methods of surveying vegetation on a large scale." New Phytologist 3.9 (1904): 228-237.

25. Praveena, R., V. Sowmya, and K. P. Soman. "Least Square Based Signal Denoising Using Wavelet Filters."

26. Mohan, Neethu, and K. P. Soman. "Power system frequency and amplitude estimation using variational mode decomposition and chebfun approximation system." In 2018 twenty fourth national conference on communications (NCC), pp. 1-6. IEEE, 2018.

27. Sujadevi, V. G., K. P. Soman, S. Sachin Kumar, Neethu Mohan, and A. S. Arunjith. "Denoising of phonocardiogram signals using variational mode decomposition." In 2017 international conference on advances in computing, communications and informatics (ICACCI), pp. 1443-1446. IEEE, 2017.

28. Mohan, Neethu, Sachin Kumar, Prabaharan Poornachandran, and K. P. Soman. "Modified variational mode decomposition for power line interference removal in ECG signals." International Journal of Electrical and Computer Engineering 6, no. 1 (2016): 151.

29. Premjith, B., K. P. Soman, M. Anand Kumar, and D. Jyothi Ratnam. "Embedding linguistic features in word embedding for preposition sense disambiguation in English—Malayalam machine translation context." In Recent Advances in Computational Intelligence, pp. 341-370. Springer, Cham, 2019.

30. Haddi, Emma, Xiaohui Liu, and Yong Shi. "The role of text pre-processing in sentiment analysis." Procedia computer science 17 (2013): 26-32.

31. Wang, Bin, Angela Wang, Fenxiao Chen, Yuncheng Wang, and C-C. Jay Kuo. "Evaluating word embedding models: methods and experimental results." APSIPA transactions on signal and information processing 8 (2019).

32. Lee, Daniel D., P. Pham, Y. Largman, and A. Ng. Advances in neural information processing systems 22. Tech. Rep., Tech. Rep, 2009.

33. Chakravarthi, Bharathi Raja, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John P. McCrae. "A sentiment analysis dataset for code-mixed Malayalam-English." arXiv preprint arXiv:2006.00210 (2020).

34. Chakravarthi, Bharathi Raja, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John P. McCrae. "Corpus creation for sentiment analysis in code-mixed Tamil-English text." arXiv preprint arXiv:2006.00206 (2020).

35. Hande, Adeep, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. "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, pp. 54-63. 2020.

36. Jelodar, Hamed, Yongli Wang, Mahdi Rabbani, Sajjad Bagheri Baba Ahmadi, Lynda Boukela, Ruxin Zhao, and Raja Sohail Ahmed Larik. "A NLP framework based on meaningful latent-topic detection and sentiment analysis via fuzzy lattice reasoning on youtube comments." Multimedia Tools and Applications 80, no. 3 (2021): 4155-4181.

37. Chakravarthi, Bharathi Raja, Ruba Priyadharshini, Sajeetha Thavareesan, Dhivya Chinnappa, Durairaj Thenmozhi, Elizabeth Sherly, John P. McCrae et al. "Findings of the sentiment analysis of dravidian languages in code-mixed text." arXiv preprint arXiv:2111.09811 (2021).

38. Malykh, Valentin, Varvara Logacheva, and Taras Khakhulin. "Robust word vectors: Context-informed embeddings for noisy texts." In Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text, pp. 54-63. 2018.

39. Dowlagar, Suman, and Radhika Mamidi. "Graph convolutional networks with multi-headed attention for code-mixed sentiment analysis." In Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pp. 65-72. 2021.

40. Clark, Eleanor, and Kenji Araki. "Text normalization in social media: progress, problems and applications for a pre-processing system of casual English." Procedia-Social and Behavioral Sciences 27 (2011): 2-11.

41. Kumaresan, Prasanna Kumar, Ratnasingam Sakuntharaj, Sajeetha Thavareesan, Subalalitha Navaneethakrishnan, Anand Kumar Madasamy, Bharathi Raja Chakravarthi, and John P. McCrae. "Findings of shared task on offensive language identification in Tamil and Malayalam." In Forum for Information Retrieval Evaluation, pp. 16-18. 2021.

42. Kumar, CS Ayush, Advaith Maharana, Srinath Murali, B. Premjith, and Soman Kp. "BERT-Based Sequence Labelling Approach for Dependency Parsing in Tamil." In Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages, pp. 1-8. 2022.

43. Shanmugalingam, Kasthuri, Sagara Sumathipala, and Chinthaka Premachandra. "Word level language identification of code mixing text in social media using nlp." In 2018 3rd International Conference on Information Technology Research (ICITR), pp. 1-5. IEEE, 2018.

44. Rane, Ankita, and Anand Kumar. "Sentiment classification system of Twitter data for US airline service analysis." In 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), vol. 1, pp. 769-773. IEEE, 2018.

45. Gopika, P., V. Sowmya, E. A. Gopalakrishnan, and K. P. Soman. "Performance improvement of deep learning architectures for phonocardiogram signal classification using fast fourier transform." In 2019 9th international conference on advances in computing and communication (ICACC), pp. 290-294. IEEE, 2019.

List of Publications based on this research work

1. Pavan Kumar P.H.V, Premjith B, Sanjanasri J.P, Soman K.P *Deep Learning Based Sentiment Analysis for Malayalam,Tamil and Kannada Languages*, FIRE 2021 Forum for Information Retrieval Evaluation Virtual Event.