Telugu News Classification using N-grams

and Bi-LSTM and Bert

# ABSTRACT

Telugu news classification aims to automatically classify Telugu news articles into several topics and is a crucial task in the field of natural language processing and information retrieval. The goal of this study is to create classification models that can organise and analyse Telugu news content automatically. The study begins by pre-processing the Telugu news data, including sentence tokenization and normalization. The N-gram approach is then employed to build a vocabulary of N-grams from the pre-processed text. The frequency of each N-gram is calculated, providing valuable insights into the importance and relevance of specific word sequences.

For Telugu news classification, the study investigates various machine learning and deep learning techniques. The Telugu news data must first go through a pre-processing process that includes text normalisation, sentence splitting, and tokenization. The Telugu language's linguistic patterns and semantic data are extracted using feature extraction techniques like N-grams and word embeddings. The accuracy with which various classification algorithms, such as Naive Bayes deep learning models like Long short-term memory , can classify Telugu news articles is examined. The models' performance is measured using performance evaluation metrics like accuracy, precision, recall, and F1-score.

Keywords— Deep Learning, Naïve Bayes, LSTM, Telugu News Classification, Evaluation Metrices, N gram

# INTRODUCTION

Telugu, one of the major Dravidian languages spoken in India, has a rich and vibrant media landscape with a substantial volume of news articles published daily. With the rapid growth of digital media and the increasing availability of Telugu news content, the need for efficient organization and classification of these articles has become crucial. Telugu news classification, the process of automatically categorizing Telugu news articles into different topics, plays a pivotal role in information retrieval, content recommendation, and data analysis.

The classification of Telugu news articles presents unique challenges due to the linguistic characteristics and complexity of the Telugu language. Telugu is an agglutinative language with a complex grammar, extensive vocabulary, and diverse sentence structures. Moreover, Telugu news articles cover a wide range of topics, including politics, entertainment, sports, technology, and more. Effectively classifying these articles into relevant topics requires the utilization of robust and language-specific classification techniques.

In this study, we use a Bi-LSTM neural network to build a reliable Telugu news classification model. To prepare the Telugu news articles for input into the Bi-LSTM model, we use a preprocessing step that tokenizes them. The model is subsequently trained using a labelled dataset of Telugu news articles, each of which is assigned a particular topic. The model gains the ability to extract pertinent features and patterns from the data during training and generalize this understanding to make predictions about articles it hasn't yet seen.

The results of this study advance Telugu news classification methods by demonstrating the effectiveness of Bi-LSTM neural networks in correctly classifying Telugu news content. The application of this model enables effective Telugu news recommendations, content organization, and information retrieval.

# LITERATURE SURVEY

Theses are the literature review of related works and existing systems in the News classification usingmachine learning and Deep Learning.

The article[1] titled "A Comparative Study on Term Weighting Methods for Automated Telegu Text Categorization with Effective Classifiers" by Vishnu Murty, Dr. B. Vishnu Vardhan, K. Sarngam, and P. Vijay pal Reddy,

the authors found that the Support Vector Machine (SVM) classifier outperformed the Naive Bayes (NB) and k-Nearest Neighbors (KNN) classifiers on six different datasets, based on F1 and macro-averaged F results. In terms of term weighting methods, they observed that the TF-RF (Term Frequency-Rare Feature) scheme consistently performed significantly better for all category distributions

In the paper[2] titled "Comparative Study on Telugu Text Classification using Machine Learning and Deep Learning models" by V. Gampala, J. Vallapuneni, P. Kumar Ande, R. Kumar Indurthi, and N. Rajesh,According to their findings, the Support Vector Machine (SVM) model achieved an overall accuracy of 87%, while the Naive Bayes (NB) model achieved 92% accuracy. The neural network model obtained an accuracy of 90%. Among all the classifiers, the Health and Technology classes demonstrated similar accuracy levels

In the paper[3], the authors compare the classification methods of TF-IDF, SVM, CNN, and LSTM for news text classification. They evaluate the performance using several key indicators, including Accuracy Rate, Loss Rate, and F1 score

In the paper[4] Sultana, J., Macigi, U. R., & Priya, G. B. K. "Telugu News Data Classification using Machine Learning Approach." the authors focus on Telugu news data classification using a machine learning approach. They employ several machine learning classifiers for this task, including Naïve Bayes, Random Forest, Passive Aggressive Classifier, Support Vector Machines (SVM), and Perceptron.

The paper[5] Naga Sudha D1, Y Madhavee Latha2. "Comparison of Text Classification Models for Telugu News Articles. The models they compare include Logistic Regression, Naïve Bayes, Support Vector Machine (SVM), and Gradient Boost Tree

# DATA DESCRIPTION

The Telugu News dataset used in this study was obtained from Kaggle, specifically sourced from SRK in 2020. This dataset comprises a total of 17,312 document sentence headings in the Telugu language. The news statements included in the dataset are categorized into five interesting zones, which serve as the class labels for classification purposes. The categories or zones of interest in the dataset are business, editorial, entertainment, nation, and sports. This dataset was selected due to the significance of news as a means of conveying reality and its considerable impact on social practices. The researchers recognized the relative lack of attention given to Telugu news within the Sentiment Analysis community, which motivated the acquisition and utilization of this dataset for their study.

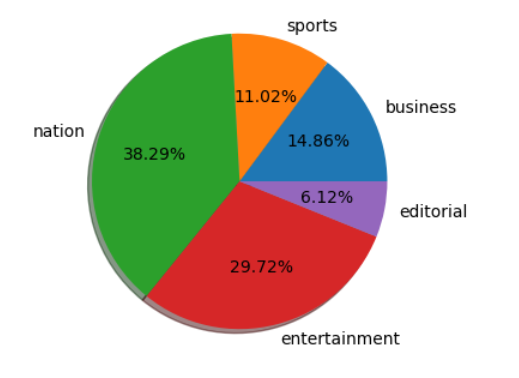


Fig 1: Count of different category of news in dataset

# METHODOLOGY

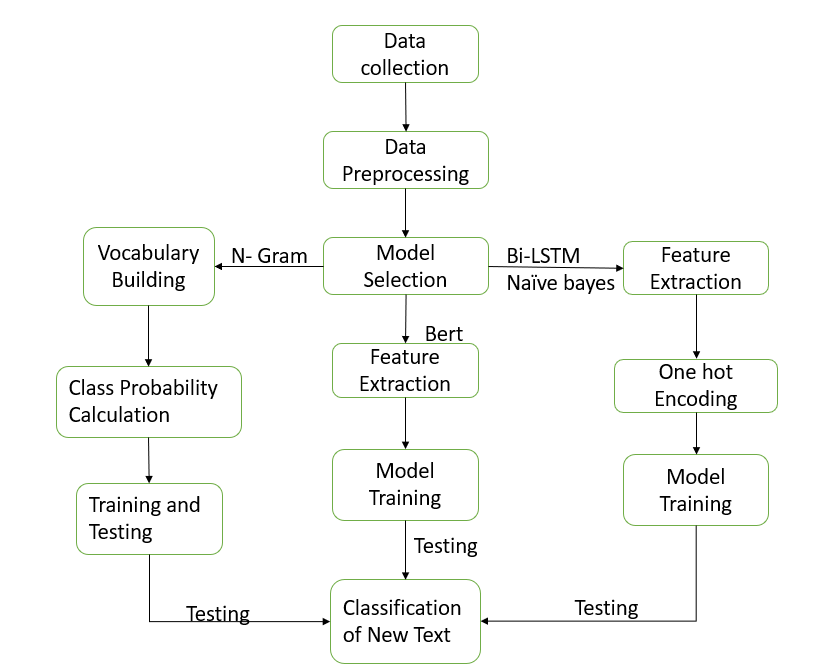


Fig2: Methodology and Flow Chart

The methodology for text classification using Naive Bayes and LSTM models with Count Vectorizer and word embeddings involves several key steps. First, the data is prepared by splitting it into training and testing sets. For Naive Bayes, the Count Vectorizer is used to convert text into numerical features, and the model is trained and evaluated for accuracy. For LSTM, the text is tokenized, padded, and the labels are one-hot encoded. The LSTM model is built using Keras and TensorFlow, trained on the training set, and evaluated on the testing set. Accuracy is calculated as a measure of performance. These steps enable the classification of Telugu news articles into predefined topics.

# NETWORK ARCHITECTURE

1. N - Gram Model

Nearby groups of n items in a text are known as n-grams. In the context of natural language processing, an n-gram refers to a sequence of n words (also known as word-level n-gram) or n characters (character-level n-gram) within a text. N-grams are useful in language modeling and text analysis tasks, including prediction and classification. By considering the occurrence and frequency of n-grams in a text, patterns, relationships, and contextual information can be captured.

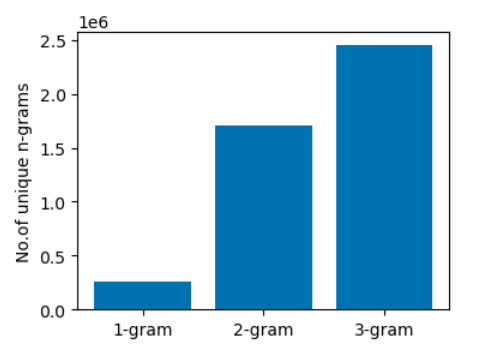


Fig 3: count of unigram, Bigram, Trigrams in data

The input data, assumed to be in a Data Frame format with 'body' and 'topic' columns, is preprocessed. Each text is converted to a string and then tokenized into sentences using a Telugu sentence tokenizer. The sentences are further tokenized into individual words using a Telugu tokenizer. N-grams, which are contiguous sequences of n words, are generated from the tokens. The n-gram vocabulary is built by counting the occurrences of each n-gram in the dataset.

For each unique class in the dataset, the class-specific data is extracted. The text in the 'body' column corresponding to each class is preprocessed, and the n-gram vocabulary is constructed. The total count of tokens in the class vocabulary is calculated. Given a new text, it is tokenized into words using the Telugu tokenizer. The probabilities of the text belonging to each class are calculated based on the n-gram vocabulary and Laplace smoothing. The probabilities are computed by considering the occurrence of each n-gram in the class vocabulary. The class with the highest probability is assigned as the predicted class for the given text.

1. Naïve Bayes

Naive Bayes is a probabilistic machine learning algorithm commonly used for text classification tasks. To classify Telugu text using Naive Bayes the Telugu text data is preprocessed by tokenizing the sentences and applying any necessary text normalization techniques The text is transformed into numerical feature vectors using techniques like the Bag-of-Words representation or TF-IDF An instance of the Naive Bayes classifier, such as Multinomial Naive Bayes, is created. The classifier is trained on a labeled dataset consisting of Telugu text samples and their corresponding class labels. After training, the classifier is used to predict the class labels of new Telugu text instances. The model calculates the probabilities of each class label given the input features and selects the label with the highest probability as the predicted class.

1. Bi-LSTM

Bidirectional LSTM (BiLSTM) is a type of recurrent neural network (RNN) architecture that is commonly used for sequential data, including text. It combines the power of both forward and backward LSTMs to capture dependencies in both directions of a sequence.

In the context of text classification, BiLSTM can be used to model the sequential nature of the input text and capture long-term dependencies. The input text is tokenized into individual words or characters. Each token is then converted into a numerical representation, such as word embeddings or character embeddings.

The sequential input is fed into a BiLSTM layer, which consists of two LSTM layers, one processing the input sequence in the forward direction and the other in the backward direction. This allows the model to capture context from both preceding and succeeding words. Additional layers can be added on top of the BiLSTM layer to further process the extracted features. This can include fully connected layers, pooling layers, or other types of recurrent or convolutional layers.

The final layer of the model is a SoftMax layer that maps the extracted features to the output classes. The SoftMax function produces probability distributions over the classes, indicating the model's confidence for each class label.

The model is trained using labeled data, where the input sequences are paired with their corresponding class labels. During training, the model learns to minimize a specified loss function, such as categorical cross-entropy. After training, the model can be used to predict the class labels of new, unseen text data.

A screenshot of a computer

Description automatically generated with medium confidence

Fig 4: Model Summary of Bi LSTM

1. Bert

BERT (Bidirectional Encoder Representations from Transformers) model for sequence classification. BERT is a state-of-the-art language model that utilizes transformer architecture to understand contextual information in text. The model specified here is based on the "Bert-base-multilingual-cased" pre-trained model, which is trained on a large corpus of text in multiple languages.

The model is designed for sequence classification tasks, where the goal is to assign a label or category to a given sequence of text. The Num labels parameter is set to 5, indicating that the model will be trained to classify input sequences into one of five different classes.

During compilation, the model is configured with the Adam optimizer, which is a popular choice for training deep learning models. The loss function used is sparse categorical cross-entropy, suitable for multi-class classification tasks. Additionally, the sparse categorical accuracy metric is employed to measure the model's performance.

This BERT model, with its ability to capture intricate contextual relationships in text, is well-suited for a wide range of natural language processing tasks, including sentiment analysis, text classification, and question answering. By fine-tuning the pre-trained BERT model on specific datasets, it can achieve remarkable results in understanding and analyzing textual data.

A screenshot of a computer program

Description automatically generated with low confidence

Fig 5: Model Summary of Bert Model

# EXPERMENTAL WORK

1. Feature Extraction

Feature extraction for an n-gram model involves preprocessing text data, generating n-grams from the tokens, and representing them numerically. Tokens are generated from the preprocessed text, and n-grams are created from these tokens. The n-grams can be represented using count-based methods, such as raw term frequencies or binary values, or by using more advanced techniques like TF-IDF. The resulting numerical representations capture the presence and frequency of n-grams in the text, enabling their use in machine learning algorithms.

For Naive Bayes with term frequency (TF) and TF-IDF, the feature extraction process involves transforming the text documents into numerical representations.

Tokenization and padding are important steps in preparing text data for a Bidirectional LSTM model. Tokenization breaks the text into individual tokens, and padding ensures all sequences are the same length. This allows the model to process the sequences effectively. The Bidirectional LSTM layer captures contextual information from past and future tokens. These steps help preprocess the data for sequence modeling and classification.

1. TRAINING

First, split the dataset into training and testing at 80% and 20 % and form that 80% training data, the data split into 75% training and 25% validation. So, training, validation, and testing sets are 60%, 20%, 20%.

1. MODELS USED

N-Grams, Naïve Bayes, Bi-LSTM

# RESULTS

1. Bi-LSTM

A picture containing text, diagram, screenshot, plot

Description automatically generated

Fig 6: Training and validation loss for Bi-LSTM

The train loss is computed by evaluating the model's performance on the training data during each training iteration. A decreasing train loss indicates that the model is effectively learning from the data and minimizing error. The declining trend in the train loss signifies that the model's ability to fit the training data is improving over time.

The validation loss is computed by evaluating the model's performance on a separate validation dataset. Similar to the train loss, a decreasing validation loss indicates that the model is becoming more accurate and generalizing better to new data. This trend suggests that the model is not overfitting, as it performs well not only on the training data but also on previously unseen data.

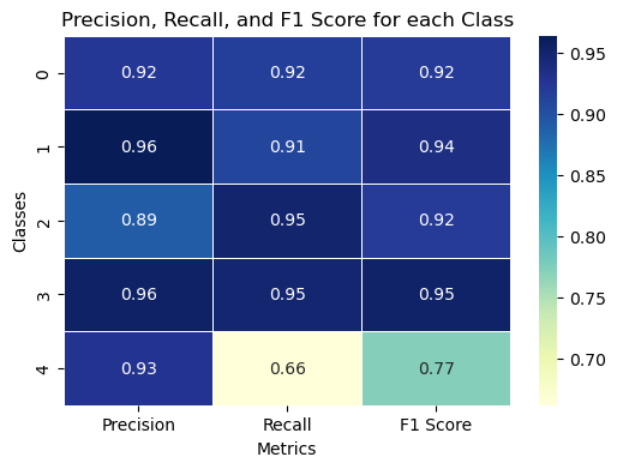


Fig7: Classification Report for Bi Lstm model

1. Naïve Bayes

The results of the Telugu text classification using Naive Bayes algorithm are as follows. The TF-IDF approach achieved an accuracy of 77%, while the TF method outperformed it with an accuracy of 87%. These accuracy scores indicate the performance of the Naive Bayes classifier in correctly categorizing Telugu texts. The TF-IDF approach considers the frequency and importance of terms in the documents, while the TF method focuses solely on term frequencies. The higher accuracy obtained with the TF method suggests that the relative frequency of terms in the documents played a significant role in the classification task. These results highlight the effectiveness of the Naive Bayes algorithm for Telugu text classification, demonstrating its ability to leverage term frequencies to accurately categorize Telugu texts.

A picture containing text, screenshot, font, number

Description automatically generated

Fig8: Classification report for Naïve bayes

A picture containing text, screenshot, diagram, square

Description automatically generated

Fig9: confusion matrix for Naïve Bayes

1. N gram model

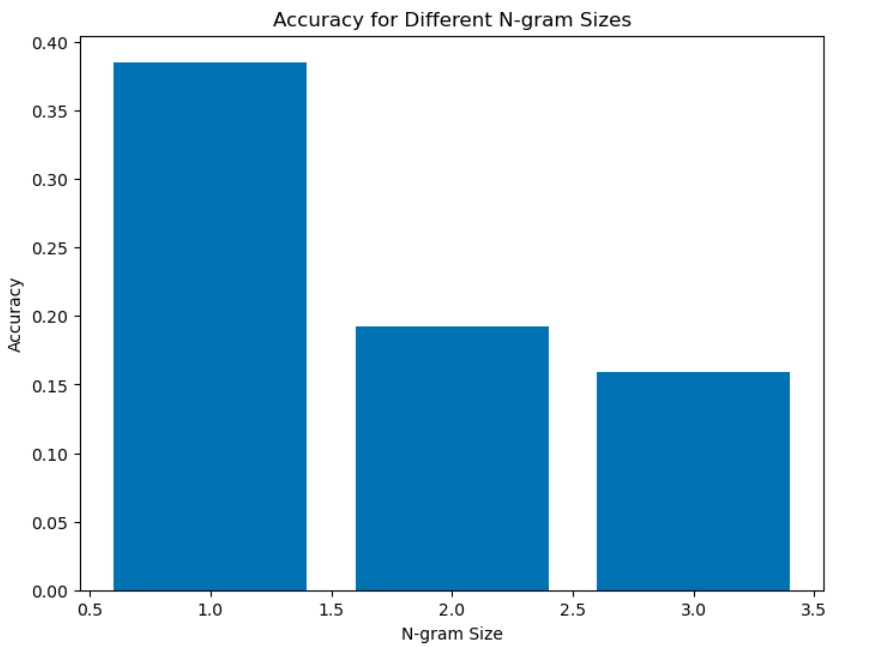


Fig9: accuracies for 1,2,3 ngrams

These results indicate that as the n-gram size increases, the accuracy of the model decreases. This is because larger n-gram sizes capture more specific and context-dependent patterns in the text. However, in the case of Telugu text classification, this increased specificity may not necessarily lead to improved performance.

Telugu language has a rich vocabulary, and when considering larger n-grams, the number of unique combinations of words increases significantly. This leads to a situation where many n-grams in the test data are unseen or have low frequencies in the training data, resulting in difficulties in accurately predicting their class labels.

1. Bert Model

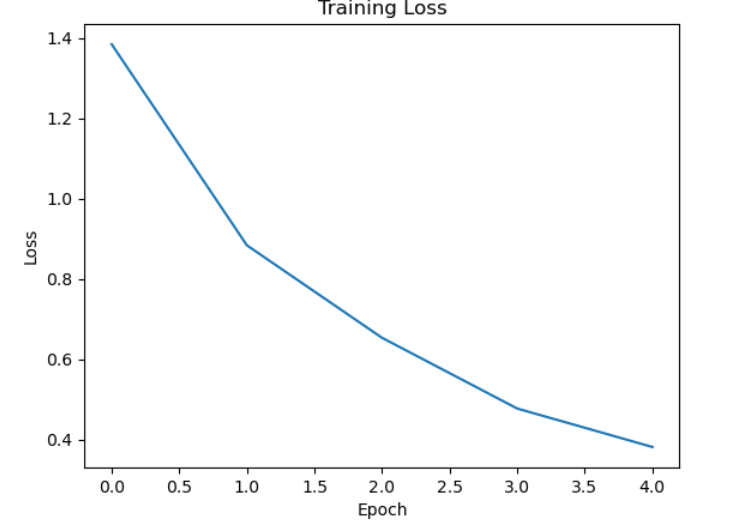


Fig10: training loss for Bert Model

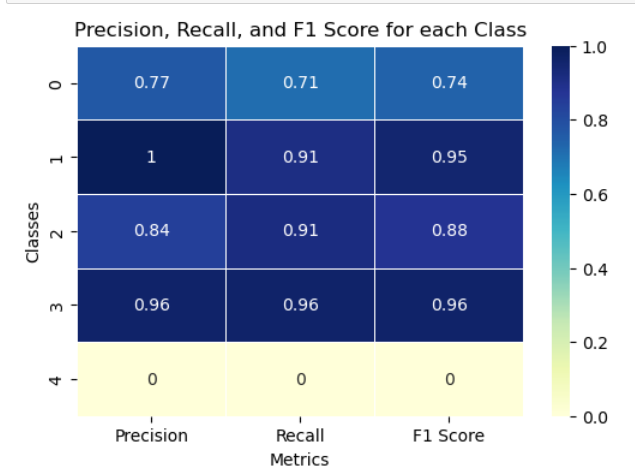


Fig 11: classification report for Bert model

In this project, a BERT (Bidirectional Encoder Representations from Transformers) model was trained for a classification task. The model demonstrated promising performance with a test loss of 0.42 and a test accuracy of 0.879.

The obtained test loss indicates that the model's predictions are relatively close to the true labels. Additionally, the test accuracy of 0.88 suggests that the BERT model predicted the correct labels for approximately 88% of the test instances. These results indicate that the model has learned meaningful representations and can generalize well to unseen data.

1. Accuracies for All models

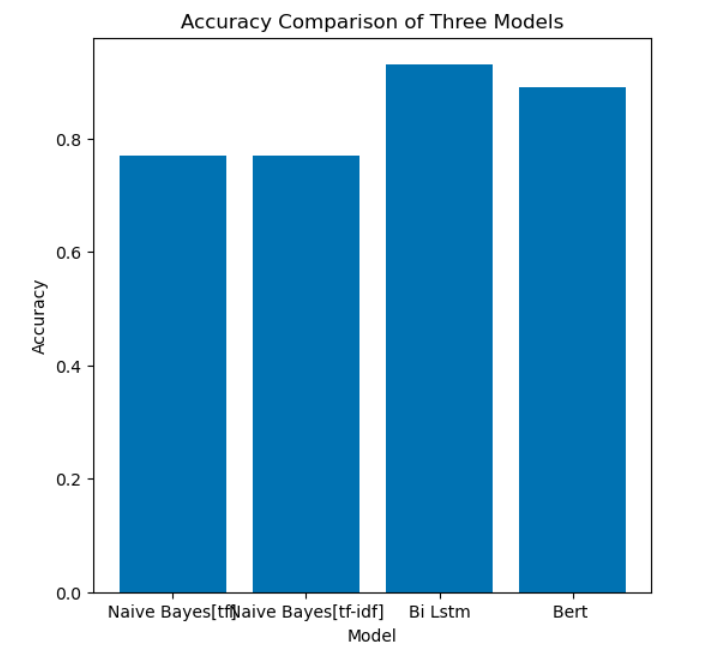


Fig12: Accuracies for Bi Lstm and naïve bayes

The Bi-LSTM model achieved a higher accuracy of 93% compared to 87% and 77% for the two features of Naive Bayes. Bi-LSTM models are capable of capturing contextual information and dependencies between words in a sequence. Naive Bayes models rely on predefined vocabularies and suffer from the issue of out-of-vocabulary (OOV) words. If a word in the test data is not present in the training vocabulary, Naive Bayes cannot assign a probability to it, resulting in limited generalization. In contrast, Bi-LSTM models can handle OOV words by learning distributed representations of words and generalizing based on the context and surrounding words.

# Conclusion

# In this study, we conducted a comparison between two models, Naive Bayes with n-gram features and a Bidirectional LSTM (Bi-LSTM) model, for Telugu text classification. The Naive Bayes model using n-grams achieved relatively low accuracy, with 38.5%, 19.2%, and 15.9% for unigrams, bigrams, and trigrams, respectively. In contrast, the Bi-LSTM model demonstrated exceptional performance, achieving an accuracy of 93%. This superior accuracy can be attributed to the Bi-LSTM model's ability to capture contextual information and handle non-linear relationships. Furthermore, when incorporating BERT and training it on the entire sequence of data, it achieved an accuracy of 87%. These results emphasize the effectiveness of deep learning models, particularly Bi-LSTM and BERT, in accurately classifying Telugu text, surpassing the performance of Naive Bayes models with n-gram features.

# Aknowledgement

I would like to express my sincere gratitude to Ms. Priyanka Vivek for supporting and helping in the successful completion of my term project. Their assistance and encouragement were invaluable throughout this journey.

# References

1. Murty, V., Vardhan, B. V., Sarngam, K., & Reddy, P. V. P. (2019). A Comparative Study on Term Weighting Methods for Automated Telugu Text Categorization with Effective Classifiers. International Journal of Data Mining and Knowledge Management Process, Volume(Issue), Page range. DOI: 10.5121
2. V. Gampala, J. Vallapuneni, P. Kumar Ande, R. Kumar Indurthi, and N. Rajesh. "Comparative Study on Telugu Text Classification using Machine Learning and Deep Learning models." In 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2021, pp. 1393-1398. DOI: 10.1109/ICOEI51242.2021.9453040.
3. C. Li, G. Zhan, and Z. Li. "News Text Classification Based on Improved Bi-LSTM-CNN." In 2018 9th International Conference on Information Technology in Medicine and Education (ITME), Hangzhou, China, 2018, pp. 890-893. DOI: 10.1109/ITME.2018.00199.
4. Sultana, J., Macigi, U. R., & Priya, G. B. K. "Telugu News Data Classification using Machine Learning Approach." In Editor's Name (Ed.), Research on Advances in Data Analytics and Complex Communication Networks (pp. XXX-XXX). Publisher: IGI Global, Year. DOI: 10.4018/978-1-7998-7685-4.ch014.
5. Naga Sudha D1, Y Madhavee Latha2. "Comparison of Text Classification Models for Telugu News Articles." In IJIRT, Volume 4, Issue 12, May 2018, ISSN: 2349-6002.
6. V. Sri Sravya, S. Kumar S and K. P. Soman, "Text Categorization of Telugu News Headlines," 2022 2nd International Conference on Intelligent Technologies (CONIT), Hubli, India, 2022, pp. 1-6, doi: 10.1109/CONIT55038.2022.9847875.
7. S. Narala, B. P. Rani and K. Ramakrishna, "Telugu text catego-rization using language models", Global journal of computer science and technology, 2017.
8. A.K. Durga and A. Govardhan, "Ontology based text categorization-telugu document", International Journal of Scientific and Engineering Research, vol. 2, no. 9, pp. 1-4, 2011
9. M. Jagadeesh, M. Anand Kumar and K.P. Soman, "Deep belief network based part-of-speech tagger for Telugu language", Proceedings of the Second International Conference on Computer and Communication Technologies, pp. 75-84, 2016
10. V.K. Vijayan, K.R. Bindu and L. Parameswaran, "A comprehensive study of text classification algorithms", 2017 Inter-national Conference on Advances in Computing Communications and Informatics (ICACCI), pp. 1109-1113, September 2017