SENTIMENT ANALYSIS OF MOVIE REVIEWS IN TAMIL LANGUAGE USING HYBRID DL MODELS

A PROJECT REPORT

Submitted by

VIJI M (20IT119)

MADHUMITHA (20IT048)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

THIAGARAJAR COLLEGE OFENGINEERING, MADURAI–15

(A Govt. Aided, Autonomous Institution, Affiliated to Anna University)

ANNAUNIVERSITY: CHENNAI 600025

May 2024

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI-15

(A Govt. Aided, Autonomous Institution, Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this project report **"SENTIMENT ANALYSIS OF MOVIE REVIEWS IN TAMIL LANGUAGE USING HYBRID DL MODELS"** is the Bonafide work of **"VIJI M(20IT119) MADHUMITHA T(20IT048)"**who carried out the project work under my supervision during the Academic Year 2023-2024.

SIGNATURE

DR. C. DEISY

PROFESSOR & HEAD

HEAD OF THE DEPARTMENT

Information Technology,
Thiagarajar College of Engineering,
Tiruparankundram,
Madurai – 625 015

SIGNATURE

MS. C. V. NISHA ANGELINE

ASSISTANT PROFESSOR

SUPERVISOR

Information Technology,
Thiagarajar College of Engineering,
Tiruparankundram,
Madurai – 625 015

Submitted for the VIVAVOCE Examination held at Thiagarajar College of Engineering on _____

INTERNAL EXAMINER **EXTERNALEXAMINER**

ACKNOWLEDGEMENT

We express our sincere gratitude to our Honorable Correspondent **Mr.K. Hari Thiagarajan** for facilitating with a good learning environment to improve our academic knowledge and performance at this premier institution.

We wish to express our profound gratitude to our beloved Principal **Dr.M.Palaninatharaja** for his overwhelming support provided during our course span in this institution.

We wish to express the gratitude to our beloved Head of the Department of Information Technology **Dr.C.Deisy,** for her motivation and support during the course span. We are grateful to our project supervisor **Ms. C. V. Nisha angeline**, for given the guidance to complete the project and led us from the scratch of this project's commencement with undivided attention and skillful expertise in developing this project in a systematic and professional manner.

We wish to express our thanks to the project review members and other faculty members of Department of Information Technology for their valuable suggestions and encouragement periodically. We express our sincere thanks to all the lab technicians for their constant and dedicated services to help us all the time.

We thank God Almighty, our parents and friends for helping us to work in a challenging, an interesting project, and in completing the same in due course without much difficulty.

VIJI M (20IT119)

MADHUMITHA(20IT048)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVATION	FULL FORM
BiLSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional NeuralNetwork
GRU	The Gated Recurrent Unit
NLP	Natural Language Processing

ABSTRACT

We conducted sentiment analysis on a dataset of Movie reviews which are available in Tamil language using both rule-based and deep learning-based approaches. The dataset consists of reviews from various Tamil movie reviews, spanning different categories and star ratings. First, we preprocessed the text data, removing noise and irrelevant characters, and then applied various natural language processing techniques such as tokenization, lemmatization, and stop word removal. we explored sentiment analysis of Tamil movie reviews using various deep learning (DL) models. Sentiment analysis involves determining the sentiment expressed in text data, whether it's positive, negative. With the increasing popularity of Tamil cinema, understanding audience sentiment towards movies becomes crucial for filmmakers and producers. Our objective is to develop an efficient system that automatically analyzes the sentiment of Tamil movie reviews written in Tamil language. To achieve this goal, we employ several DL models such as recurrent neural networks (RNNs), convolutional neural networks (CNNs). After training, the models are evaluated using standard evaluation metrics to assess their performance in sentiment classification. We compare the results obtained from different DL models to identify the most effective approach for sentiment analysis of Tamil movie reviews. Furthermore, to make our sentiment analysis accessible and user-friendly, we integrate the trained model into a website. Users can input Tamil movie reviews, and our system will provide sentiment analysis output, indicating whether the review expresses positive, negative, or neutral sentiment. The website interface enhances the usability and accessibility of our sentiment analysis tool, catering to Tamil cinema enthusiasts, filmmakers, and industry professionals.

Keywords: CNN,RNN,DL, NLP

CHAPTER 1
INTRODUCTION

With value for filmmakers, producers, and enthusiasts alike. With the surge in Tamil cinema's global appeal, the need to comprehend the sentiments expressed in Tamil movie reviews becomes paramount. This project delves into the realm of sentiment analysis, focusing specifically on Tamil movie reviews written in the Tamil language. Leveraging the power of deep learning (DL) models, we aim to construct a robust system capable of automatically analyzing and categorizing the sentiment conveyed within these reviews.

Sentiment analysis, a subset of natural language processing (NLP), entails the task of discerning the underlying sentiment—whether positive, negative, or neutral—from textual data. Through sentiment analysis, stakeholders in the film industry gain insights into audience reactions, enabling them to make informed decisions regarding movie production, marketing strategies, and audience engagement.

In this endeavor, we embark on a journey to explore the efficacy of various DL models in the context of Tamil sentiment analysis. We delve into the intricacies of recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer-based architectures , adapting them to the nuances of the Tamil language.

Our methodology involves training these DL models on a meticulously curated dataset comprising annotated Tamil movie reviews. Through meticulous preprocessing techniques including tokenization and embedding, we prepare the textual data for ingestion by the DL models, ensuring optimal performance and accuracy.

Following the training phase, we subject the models to rigorous evaluation, employing standard metrics to gauge their effectiveness in sentiment classification.

By comparing and contrasting the results obtained from different DL architectures, we aim to identify the most proficient approach for Tamil sentiment analysis.

Moreover, to enhance the accessibility and usability of our sentiment analysis solution, we integrate the trained model into a user-friendly website interface. This interface empowers users to input Tamil movie reviews effortlessly, receiving instantaneous sentiment analysis output—whether the review exudes positivity, negativity, or neutrality.

Through this project, we aspire to contribute to the burgeoning field of NLP by tailoring a sentiment analysis solution specifically for Tamil movie reviews. By harnessing the capabilities of DL models and encapsulating them within a user-friendly website, we endeavor to facilitate informed decision-making within the Tamil cinema ecosystem while fostering greater engagement and interaction among enthusiasts. In this project, we embark on a journey to harness the power of deep learning (DL) models for sentiment analysis of Tamil movie reviews. Our goal is to develop a sophisticated system capable of accurately categorizing reviews into positive, negative, or neutral sentiments, all while preserving the nuances of the Tamil language.

We begin by assembling a comprehensive dataset of annotated Tamil movie reviews, meticulously curated to capture the diversity of opinions prevalent within the Tamil cinema community. This dataset serves as the foundation upon which we train and evaluate a variety of DL models, including recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer-based

Through rigorous experimentation and fine-tuning, we seek to identify the most effective DL model for Tamil sentiment analysis, considering factors such as accuracy, efficiency, and scalability. Our methodology involves preprocessing the textual data to ensure compatibility with the chosen DL architectures, employing techniques such as tokenization and word embedding.

Once trained and validated, the selected DL model is seamlessly integrated into a user-friendly website interface, democratizing access to sentiment analysis for

Tamil movie reviews. Users can effortlessly input their reviews and receive instant feedback on the sentiment expressed, empowering them to engage with Tamil cinema in a more informed manner.

By undertaking this project, we aim to not only advance the state-of-the-art in NLP but also to contribute a valuable tool to the Tamil cinema ecosystem. Our sentiment analysis solution promises to facilitate deeper insights, foster meaningful dialogue, and ultimately enrich the cinematic experience for audiences and creators alike.

he gap between advanced computational techniques and the rich tapestry of Tamil cinema. As we delve into the intricacies of sentiment analysis, we acknowledge the unique linguistic nuances and cultural contexts that shape the Tamil language and its cinematic expressions.

Our endeavor is driven by a dual commitment—to enhance the efficiency and accuracy of sentiment analysis through DL models, and to celebrate the diversity and vibrancy of Tamil cinema. By leveraging the latest advancements in artificial intelligence and machine learning, we aspire to unlock deeper insights into audience sentiments, empowering stakeholders to navigate the ever-evolving landscape of Tamil cinema with clarity and confidence.

In the following sections, we delve into the methodology, experimentation, and results of our sentiment analysis project, offering a comprehensive overview of our approach and findings. Through meticulous research and experimentation, we aim to illuminate the potential of DL models in deciphering the complex emotions and opinions encapsulated within Tamil movie reviews.

Ultimately, our journey transcends the realm of technology—it is a celebration of language, culture, and storytelling. By harnessing the power of DL models to analyze Tamil sentiment, we endeavor to enrich the cinematic experience for audiences, creators, and industry professionals alike, fostering a deeper appreciation and understanding of Tamil cinema's enduring legacy.

1.1 OBJECTIVES AND CONSTARINTS

OBJECTIVES:

- Develop a robust sentiment analysis system tailored specifically for Tamil movie reviews, leveraging deep learning (DL) models.
- Curate a comprehensive dataset of annotated Tamil movie reviews, encompassing diverse sentiments and linguistic variations.
- Explore and evaluate various DL architectures, including recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer-based models such as BERT, to identify the most effective approach for Tamil sentiment analysis.
- Preprocess the textual data using techniques such as tokenization and word embedding to ensure compatibility with the chosen DL models.
- Train and fine-tune the DL models on the annotated dataset, optimizing for accuracy, efficiency, and scalability.
- Conduct rigorous evaluation of the trained models using standard metrics to assess their performance in sentiment classification.
- Integrate the selected DL model into a user-friendly website interface, enabling users to input Tamil movie reviews and receive instant sentiment analysis output.
- Enhance the accessibility and usability of the sentiment analysis tool, catering to Tamil cinema enthusiasts, filmmakers, and industry professionals.
- Provide insights and recommendations based on the sentiment analysis results to aid decision-making processes within the Tamil cinema ecosystem.
- Contribute to the advancement of natural language processing (NLP) research by developing a specialized solution for sentiment analysis of Tamil textual

data, paving the way for future innovations in linguistic analysis

CONSTRAINTS:

- **Language Specificity:** The sentiment analysis system is designed specifically for Tamil movie reviews written in the Tamil language. Adapting the system to analyze reviews in other languages would require significant additional development and resources.
- **Data Availability:** The availability of annotated Tamil movie review datasets may be limited, posing a constraint on the quantity and diversity of training data for the DL models. This could impact the generalization and performance of the sentiment analysis system.
- **Model Interpretability**: Deep learning models, particularly complex architectures like transformers, often lack interpretability, making it challenging to understand the reasoning behind their predictions. Ensuring transparency and interpretability of the sentiment analysis system could be a constraint in model selection and deployment.
- **Performance Metrics**: Evaluating the performance of the sentiment analysis system relies on standard metrics such as accuracy, precision, recall, and F1-score. However, these metrics may not fully capture the nuances of sentiment analysis, especially in multiclass classification tasks with imbalanced datasets.
- **User Interface Design**: Designing a user-friendly website interface for inputting Tamil movie reviews and displaying sentiment analysis output requires consideration of usability principles, accessibility guidelines, and cultural preferences. Constraints in user interface design may affect the user experience and adoption of the system.

1.1 OBJECTIVES AND CONSTARINTS

OBJECTIVES:

7 Develop a robust sentiment analysis system tailored specifically for Tamil movie reviews, leveraging deep learning (DL) models.
8 Curate a comprehensive dataset of annotated Tamil movie reviews, encompassing diverse sentiments and linguistic variations.
9 Explore and evaluate various DL architectures, including recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer-based models such as BERT, to identify the most effective approach for Tamil sentiment analysis.
10 Preprocess the textual data using techniques such as tokenization and word embedding to ensure compatibility with the chosen DL models.
11 Train and fine-tune the DL models on the annotated dataset, optimizing for accuracy, efficiency, and scalability.
12 Conduct rigorous evaluation of the trained models using standard metrics to assess their performance in sentiment classification.
13 Integrate the selected DL model into a user-friendly website interface, enabling users to input Tamil movie reviews and receive instant sentiment analysis output.
14 Enhance the accessibility and usability of the sentiment analysis tool, catering to Tamil cinema enthusiasts, filmmakers, and industry professionals.
15 Provide insights and recommendations based on the sentiment analysis results to aid decision-making processes within the Tamil cinema ecosystem.
16 Contribute to the advancement of natural language processing (NLP) research by developing a specialized solution for sentiment analysis of Tamil textual

data, paving the way for future innovations in linguistic analysis

CONSTRAINTS:

1. **Language Specificity:** The sentiment analysis system is designed specifically for Tamil movie reviews written in the Tamil language. Adapting the system to analyze reviews in other languages would require significant additional development and resources.
2. **Data Availability:** The availability of annotated Tamil movie review datasets may be limited, posing a constraint on the quantity and diversity of training data for the DL models. This could impact the generalization and performance of the sentiment analysis system.
3. **Model Interpretability**: Deep learning models, particularly complex architectures like transformers, often lack interpretability, making it challenging to understand the reasoning behind their predictions. Ensuring transparency and interpretability of the sentiment analysis system could be a constraint in model selection and deployment.
4. **Performance Metrics**: Evaluating the performance of the sentiment analysis system relies on standard metrics such as accuracy, precision, recall, and F1-score. However, these metrics may not fully capture the nuances of sentiment analysis, especially in multiclass classification tasks with imbalanced datasets.
5. **User Interface Design**: Designing a user-friendly website interface for inputting Tamil movie reviews and displaying sentiment analysis output requires consideration of usability principles, accessibility guidelines, and cultural preferences. Constraints in user interface design may affect the user experience and adoption of the system.

- **USED DL MODELS**

1.**CNN_LSTM:**

The CNN-LSTM (Convolutional Neural Network - Long Short-Term Memory) architecture combines the strengths of both CNNs and LSTMs to capture spatial and temporal dependencies in sequential data like text. Here's an overview of how CNN-LSTM works:

1. **Convolutional Neural Network (CNN):**
    - CNNs are well-known for their effectiveness in extracting spatial features from input data.
    - In the context of text data, the CNN component of the architecture typically operates on word embeddings, treating them as 1D sequences.
    - The CNN layer applies a series of convolutional filters to the input sequence, capturing local patterns or features.
    - Max-pooling or global pooling operations are often employed to reduce the dimensionality of the feature maps while retaining the most relevant information.
2. **Long Short-Term Memory (LSTM):**
    - LSTMs are a type of recurrent neural network (RNN) designed to model temporal dependencies in sequential data.
    - They excel at capturing long-range dependencies and mitigating the vanishing gradient problem often encountered in traditional RNNs.
    - LSTMs consist of memory cells with self-gating mechanisms, including input gates, forget gates, and output gates, allowing them to retain and selectively update information over time.
3. **Integration of CNN and LSTM:**
    - In the CNN-LSTM architecture, the output of the CNN layer serves as input to the LSTM layer.

- The CNN layer acts as a feature extractor, capturing high-level spatial features from the input sequence.
- These extracted features are then fed into the LSTM layer, which models the temporal dynamics of the sequence, incorporating both short-term and long-term dependencies.

4. **Training and Optimization:**
    - The CNN-LSTM architecture is trained end-to-end using backpropagation through time (BPTT) or similar optimization algorithms.
    - During training, the parameters of both the CNN and LSTM components are updated iteratively to minimize a predefined loss function, often associated with a specific task such as sentiment analysis or sequence classification.
    - Hyperparameter tuning, including the number of convolutional layers, filter sizes, LSTM units, and dropout rates, plays a crucial role in optimizing the performance of the CNN-LSTM model.
5. **Applications:**
    - CNN-LSTM architectures have been successfully applied to various sequential data tasks, including natural language processing (NLP), speech recognition, video analysis, and time series forecasting.
    - In NLP, CNN-LSTM models are commonly used for tasks such as sentiment analysis, text classification, machine translation, and named entity recognition, leveraging their ability to capture both spatial and temporal information in textual data.

ACCURACY: From the CNN_LSTM model we got the accuracy of 74 percent From training the model

2.CNN-BILSTM:

The CNN-BI LSTM (Convolutional Neural Network - Bidirectional Long Short-Term Memory) architecture is a hybrid neural network model that combines the strengths of both CNNs and bidirectional LSTMs to capture spatial features and bidirectional temporal dependencies in sequential data. Here's an overview of how CNN-BI LSTM works:

1. **Convolutional Neural Network (CNN):**
    - CNNs are adept at extracting spatial features from input data using convolutional filters.
    - In the context of text data, the CNN component operates on word embeddings, treating them as 1D sequences.
    - Convolutional layers apply a series of filters across the input sequence to capture local patterns or features.
2. **Bidirectional Long Short-Term Memory (BI LSTM):**
    - LSTMs are a type of recurrent neural network (RNN) designed to model temporal dependencies in sequential data.
    - Bidirectional LSTMs extend the capabilities of traditional LSTMs by processing input sequences in both forward and backward directions.
    - This bidirectional processing allows the model to capture dependencies from both past and future contexts, enabling a more comprehensive understanding of the sequential data.
3. **Integration of CNN and BI LSTM:**
    - In the CNN-BI LSTM architecture, the output of the CNN layer serves as input to the bidirectional LSTM layer.
    - The CNN layer acts as a feature extractor, capturing high-level spatial features from the input sequence.
    - These extracted features are then fed into the bidirectional LSTM layer, which models the bidirectional temporal dynamics of the sequence, incorporating information from both past and future

contexts.

4. **Training and Optimization:**
    - The CNN-BI LSTM model is trained end-to-end using backpropagation through time (BPTT) or similar optimization algorithms.
    - During training, the parameters of both the CNN and bidirectional LSTM components are updated iteratively to minimize a predefined loss function, often associated with tasks such as sentiment analysis or sequence classification.
    - Hyperparameter tuning, including the number of convolutional layers, filter sizes, LSTM units, and dropout rates, is essential for optimizing the performance of the CNN-BI LSTM model.
5. **Applications:**
    - CNN-BI LSTM architectures are widely used in natural language processing (NLP) tasks such as sentiment analysis, text classification, machine translation, and named entity recognition.
    - The model's ability to capture both spatial features and bidirectional temporal dependencies makes it well-suited for tasks involving sequential data with complex patterns and long-range dependencies.

ACCURACY: From the CNN_BILSTM model we got the accuracy of 80 percent From training the model

3.**CNN_BIGRU:**

The CNN-BI GRU (Convolutional Neural Network - Bidirectional Gated Recurrent Unit) architecture combines the capabilities of CNNs and bidirectional GRUs to capture spatial features and bidirectional temporal dependencies in sequential data. Here's an overview of how CNN-BI GRU works:

1. **Convolutional Neural Network (CNN):**
    - CNNs are proficient at extracting spatial features from input data using

convolutional filters.

- In the context of text data, the CNN component treats word embeddings as 1D sequences.
- Convolutional layers apply filters across the input sequence to capture local patterns or features, preserving spatial information**.**

2. **Bidirectional Gated Recurrent Unit (BI GRU):**
    - GRUs (Gated Recurrent Units) are a type of recurrent neural network (RNN) designed to model temporal dependencies in sequential data.
    - Bidirectional GRUs enhance the traditional GRU architecture by processing input sequences in both forward and backward directions.
    - This bidirectional processing enables the model to capture dependencies from both past and future contexts, facilitating a more comprehensive understanding of the sequential data.
3. **Integration of CNN and BI GRU:**
    - In the CNN-BI GRU architecture, the output of the CNN layer serves as input to the bidirectional GRU layer.
    - The CNN layer acts as a feature extractor, extracting spatial features from the input sequence.
    - These features are then fed into the bidirectional GRU layer, which models the bidirectional temporal dynamics of the sequence, incorporating information from both past and future contexts.
4. **Training and Optimization:**
    - The CNN-BI GRU model is trained end-to-end using optimization algorithms such as backpropagation through time (BPTT).
    - During training, the parameters of both the CNN and bidirectional GRU components are updated iteratively to minimize a predefined loss function, typically associated with tasks like sentiment analysis or sequence classification.
    - Hyperparameter tuning, including the number of convolutional layers, filter sizes, GRU units, and dropout rates, is crucial for optimizing the model's performance.
5. **Applications:**

- CNN-BI GRU architectures find applications in various natural language processing (NLP) tasks such as sentiment analysis, text classification, machine translation, and named entity recognition.
- The model's ability to capture both spatial features and bidirectional temporal dependencies makes it well-suited for tasks involving sequential data with complex patterns and long-range dependencies.

ACCURACY: From the CNN_BILSTM model we got the accuracy of 79 percent
From training the model

1.3 CHOSEN MODEL

DL models are chosen for the projects based on their performance and their accuracies
Here we have totally three DL models each having its own uniqueness and quailities in
Giving the accurate predictions and we have evaluated this various models based on the
Level of Performance and we have had compared them out of their complexity and had
Chosen the best model for our project building
From the above three models(CNN_LSTM,CNN_BILSTM,CNN_BIGRU) we see that
CNN_BILSTM performs comparatively better than other two models in both performance
And accuracy vise

CHAPTER 2

LITERATURE SURVEY

Suba Sri Ramesh Babu.(14th August 2022) Sentiment Analysis In Tamil Language Using Hybrid Deep Learning Approach National College of Ireland. This research work aims to propose hybrid deep learning approaches such as CNN-BiLSTM, CNN-LSTM, and CNN-BiGRU. The proposed methods will be evaluated and compared on various metrics to find the best performing model among them. The result shows that CNN-BiLSTM has achieved the higher accuracy of 80.2% and highest f1-score of 0.64 when compared to other two models [1]

Sajeetha Thavareesan ,Sinnathamby Mahesan.(2018) REVIEW ON SENTIMENT ANALYSIS IN TAMIL TEXTS published a study in the Eastern University, Sri Lanka.This review paper mainly focuses on analysing the recent literature on the sentiment analysis field. On analysing various papers it is found that SVM and RNN classifiers taking TF-IDF and word2vec gives better performance than grammar rules based classification and other various classifiers [2]

N.Sripriya S. Dhivya.(Dec 13 2021). **Sentimental analysis for code-mixed tamil language CEUR Workshop Proceedings (CEUR-WS.org).** This paper elaborates a model that generates embedding representation for the text data available. This is a multiclass classification problem that generates five different labels for the data collected from YouTube comments. It is found that Random Forest classifier builds a set of decision trees from the randomly selected subgroup of training data. This classifier is more accurate and vigorous in making decisions due to the numerous decision trees involved in the process[3]

Pavan Kumar P.H.V, Premjith B, Sanjanasri J.P and Soman K.P(December 17-21, 2020) **Deep Learning Based Sentiment Analysis for Malayalam,Tamil and Kannada Languages CEUR Workshop Proceedings (CEUR-WS.org)**. This paper describes the we implementation of three different

Deep learning-based architectures for predicting various sentiments associated with the Dravidian CodeMix languages(Malayalam, Tamil, Kannada): 1. Deep Neural Network (DNN) 2. Bidirectional-Long Short Term Memory network (Bi-LSTM) 3. Convolution Neural network (CNN) combined with a Long Short Term Memory network (LSTM) The dataset used in this task is CodeMix text associated with the context of social media. The BiLSTM layer suits the classification of Tamil and Kannada corpus [4]

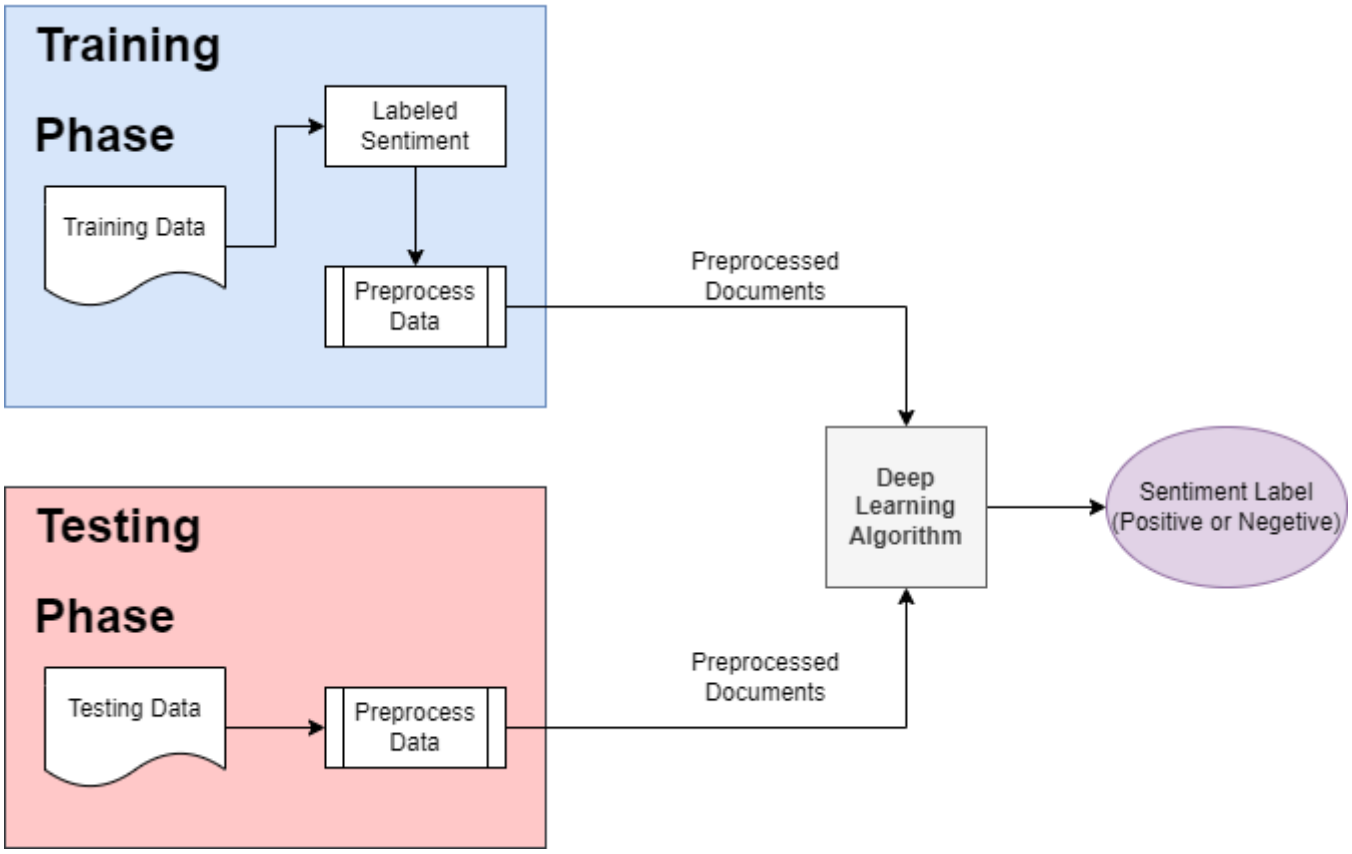
Kausikaa.N Uma.V **sentiment analysis for english and tamil tweets using path length similarity based word sense disambiguation** published in IOSR journal of computer engineering, june 2016 Here the data are collected from Twitter using Twitter Api.The collected tweets consist of mixed Tamil and English language which increases the ambiguity in classification.so for that Word Sense Disambiguation is employed to overcome this issue and semantic similarity between the words is predicted using Path based similarity. The shortest distance between two synset is found using Edge based similarity. The Support Vector Machine classifier is used to predict the polarity values of the tweets.[5]

CHAPTER 3

SYSTEM DESIGN AND IMPLEMENTATION

3.1. SYSTEM BLOCK DIAGRAM:

Figure.3.1.System diagram



1. **Training Phase:**
    - In the training phase, you start with a labeled dataset of Tamil text samples, where each sample is associated with a specific emotion label (e.g., happy, sad, angry).
    - You preprocess the text data, which may involve tasks such as tokenization, normalization, and vectorization (e.g., converting words into numerical representations).
    - Next, you split the dataset into training and validation sets. The training set is used to train the DL model, while the validation set helps monitor the model's performance during training and tune hyperparameters.
    - You select a DL model architecture suitable for the task of emotion prediction in Tamil text. This could be a recurrent neural network (RNN), convolutional neural network (CNN).
    - The selected DL model is initialized with random weights and trained iteratively using the training data. During training, the model learns to map input Tamil text samples to their corresponding emotion labels by adjusting its parameters based on the loss computed between predicted and actual labels.
    - Training involves multiple epochs (iterations over the entire training dataset), where the model updates its parameters using optimization techniques such as stochastic gradient descent (SGD) or Adam optimization.

1. **Testing Phase:**
    - Once the training phase is complete, the trained DL model is evaluated on a separate, unseen dataset called the test set.
    - Similar to the training data, the test data consists of Tamil text samples with associated emotion labels. However, the model has not seen these samples during training.
    - You preprocess the test data in the same manner as the training data to ensure consistency in data representation.
    - The DL model predicts the emotions for each Tamil text sample in the test set using its learned parameters and the inference process.
    - After making predictions, you compare the predicted emotion labels with the ground truth labels from the test set to assess the model's performance.
    - Performance metrics such as accuracy, precision, recall, and F1-score are computed to measure the model's effectiveness in predicting emotions in Tamil text.
    - The test results provide insights into the model's generalization ability and its performance on unseen data, guiding further improvements or adjustments if necessary.

DEEP LEARNING ALGORITHMS:

Hybrid Architectures:

- Hybrid architectures combine different types of neural networks, such as CNNs and RNNs, to leverage their complementary strengths.
- Examples include CNN-LSTM and CNN-BI LSTM architectures, which integrate convolutional layers for spatial feature extraction

with recurrent layers for capturing temporal dependencies.

PREDICTION

1. **Preprocessing the Input Data:**
    - Before making predictions, preprocess the unseen Tamil text samples in the same manner as the training data.
    - This preprocessing may involve tokenization (splitting the text into individual words or subwords), normalization (converting text to lowercase, removing punctuation), and encoding (converting words to numerical representations using techniques like word embeddings).
2. **Input Encoding:**
    - Convert the preprocessed Tamil text samples into numerical representations that can be understood by the trained deep learning model.
    - For models like RNNs or transformers, this typically involves representing each word or subword in the text as a vector using pre-trained word embeddings or learned embeddings from the model's embedding layer.
3. **Making Predictions:**
    - Feed the encoded Tamil text samples into the trained deep learning model to obtain predictions for the emotions expressed in those samples.
    - The model processes the input text through its layers, utilizing the learned parameters to generate predictions.
    - Depending on the architecture of the model, predictions may be generated at each time step (for sequential models like RNNs) or for the entire input sequence (for models like CNNs)

3.2. TECH STACK:

3.2.1. PROGRAMMING LANGUAGES:

The programming language used in the sentiment analysis project is Python. Python is a popular choice for natural language processing (NLP) tasks due to its extensive libraries and frameworks specifically designed for text processing and machine learning. Libraries such as NLTK (Natural Language Toolkit), spaCy, and Transformers provide powerful tools and pre-trained models for tasks like tokenization, sentiment analysis, and language modeling, making Python an ideal choice for developing NLP applications like sentiment analysis systems.

Additionally, Python's simplicity, readability, and large community support makeit well-suited for developing and maintaining complex NLP projects.

3.2.2.PACKAGES USED:

The sentiment analysis project uses several Python packages for various tasks. Basedon the provided code, here are the packages used:

NLTK (Natural Language Toolkit) stands as a robust Python library for NLP tasks, including tokenization, lemmatization, and sentiment analysis. It boasts a rich collection of corpora and lexical resources, making it a go-to tool for linguistic research and NLP development. With its comprehensive functionalities, NLTK facilitates tasks such as part-of-speech tagging, parsing, and named entity recognition, empowering users to delve deep into natural language processing tasks. Its open-source nature encourages community contributions and enhancements, ensuring its relevance in the evolving landscape of NLP technologies. NLTK's versatility extends to educational purposes, serving as an invaluable resource for teaching .

GLOVE (Global Vectors for Word Representation) embedding is a method for representing words as dense vectors in a continuous vector space. Unlike traditional one-hot encoding, where each word is represented as a sparse vector with only one non-zero element, GloVe embeddings capture semantic relationships between words by encoding them as vectors of real numbers. The fundamental concept underlying GloVe is the representation of words as vectors in a continuous vector space, where the angle and direction of the vectors correspond to the semantic connections between the appropriate words. To do this, GloVe builds a co-occurrence matrix using word pairs and then optimizes the word vectors to minimize the difference between the pointwise mutual information of the corresponding words and the dot product of vectors.

NumPy serves as the cornerstone of scientific computing in Python, offering powerful array manipulation capabilities and mathematical functions. Its multidimensional array object facilitates efficient storage and manipulation of large datasets, enabling fast numerical computations and simulations. NumPy's extensive library of mathematical functions and linear algebra routines empowers users to perform complex calculations with ease, making it indispensable for scientific research and engineering applications. Beyond numerical operations, NumPy supports advanced array manipulation techniques such as slicing, indexing, and broadcasting, enhancing productivity and code readability. Its seamless integration with other scientific libraries like SciPy and Matplotlib fosters a rich ecosystem for scientific computing and data analysis. NumPy's open-source nature encourages community contributions and enhancements, ensuring its continued relevance and adaptability to emerging computational challenges. Whether tackling mathematical problems or processing large datasets, NumPy remains a fundamental tool for Python developers and researchers alike.

Pandas emerges as a versatile and powerful library for data manipulation and analysis in Python, catering to a wide range of data processing tasks. Its DataFrame object provides a tabular data structure ideal for organizing, cleaning, and analyzing structured data. With intuitive indexing and labeling capabilities, Pandas simplifies data manipulation tasks such as filtering, sorting, and grouping, enhancing productivity for data analysts and scientists. Beyond data wrangling, Pandas offers robust functionality for statistical analysis, including descriptive statistics, correlation analysis, and time series analysis, enabling deeper insights into data patterns and trends. Its seamless integration with other Python libraries like Numpy

and vibrant community support foster learning and collaboration, empowering users to harness the full potential of their data. From exploratory data analysis to production-level data processing pipelines, Pandas remains a cornerstone of the Python data ecosystem, driving innovation and discovery across industries and domains.

TENSORFLOW TensorFlow can be used to develop models for various tasks, including natural language processing, image recognition, handwriting recognition, and different computational-based simulations such as partial differential equation

CHAPTER 4

OBJECTIVES AND METHODOLOGY

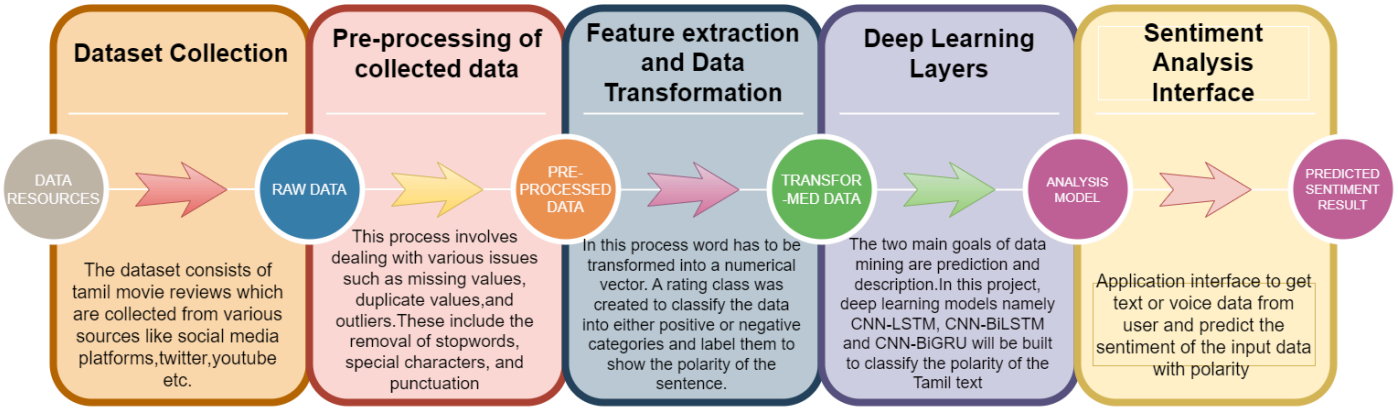
4.1. AIM:

4.2. PROJECTED OUTCOME ANALYSIS:

4.3. REQUEST FUNCTION IMPACT:

4.4. CHALLENGES IN UTILIZING MODEL:

4.5. WORKFLOW:


Dataset Collection
DATA RESOURCES
RAW DATA
The dataset consists of tamil movie reviews which are collected from various sources like social media platforms,twitter,youtube etc.
Pre-processing of collected data
PRE-PROCESSED DATA
This process involves dealing with various issues such as missing values, duplicate values,and outliers.These include the removal of stopwords, special characters, and punctuation
Feature extraction and Data Transformation
TRANSFOR-MED DATA
In this process word has to be transformed into a numerical vector. A rating class was created to classify the data into either positive or negative categories and label them to show the polarity of the sentence.
Deep Learning Layers
ANALYSIS MODEL
The two main goals of data mining are prediction and description.In this project, deep learning models namely CNN-LSTM, CNN-BiLSTM and CNN-BiGRU will be built to classify the polarity of the Tamil text
Sentiment Analysis Interface
PREDICTED SENTIMENT RESULT
Application interface to get text or voice data from user and predict the sentiment of the input data with polarity

4.6. CONSIDERATIONS:

Dataset size: The size of the dataset affects the accuracy of the model. Larger data generally produces more accurate models.

Quality of the dataset: The quality of the dataset also affects the accuracy of the model. A data set with clearer and more relevant information will result in amore accurate model.

Model complexity: Model complexity affects training time and model accuracy. More advanced models generally take longer to train but can also be more accurate.

Availability of computing resources: Availability of computing resources will affect the running time and cost of the model. More computational data will be needed to train more complex models.

4.7. SYNTHETIC PROCEDURE OF THE PROPOSED WORK:

4.8. TECHNIQUES AND ALGORITHMS:

Figure4.3.2.1.

.CHAPTER 5

IMPLEMENTATIONS

5.1. PROPOSED WORK:

5.2. PROCEDURE:

5.3. HARDWARE REQUIREMENTS:

Processor (CPU): A modern multi-core processor (e.g., Intel Core i5 or AMD Ryzen 5) should suffice for running the sentiment analysis tasks and machine learning algorithms used in the project.

Memory (RAM): At least 8 GB of RAM is recommended to handle the data preprocessing, feature extraction, and training of machine learning models efficiently. If working with large datasets, more RAM might be necessary.

Storage: Sufficient storage space is required to store the dataset, any additional

resources, and the Python environment along with its dependencies. A minimum of20 GB of free disk space should be available.

Graphics Processing Unit (GPU) (Optional): While not strictly necessary, having access to a GPU can significantly accelerate the training of machine learning models, especially deep learning models such as BERT. If you plan to use GPU acceleration, an NVIDIA GPU with CUDA support (e.g., GeForce GTX 1060 or higher) is recommended.

Operating System: The project can run on any major operating system like Windows, macOS, or Linux. Ensure that the chosen operating system is compatiblewith the required Python libraries and dependencies.

Internet Connection: A stable internet connection might be necessary for downloading additional libraries, resources, or pre-trained models, especially if youplan to use cloud-based services or external APIs.

5.4. SOFTWARE REQUIREMENTS:

1. Python: The project is developed using Python, so having Python installedon your system is essential. Python provides a wide range of libraries and tools for natural language processing (NLP) and machine learning tasks.

2. Python Libraries: Various Python libraries are used for NLP, machine learning, and data manipulation tasks. Some common libraries include:

- NLTK (Natural Language Toolkit)
- scikit-learn
- Pandas
- NumPy
-

- Hugging Face Transformers

3. Integrated Development Environment (IDE): While not strictly necessary, using an IDE such as PyCharm, Jupyter Notebook, or VS Code can facilitate developmentand debugging tasks.

4. Other Tools and Packages: Depending on the specific requirements of the project,you may need additional tools and packages for data visualization, model evaluation,or deployment.

CHAPTER 6

RESULTS AND DISCUSSION

6.1. RESULTS:

6.1.1. ADVANTAGES:

:

6.2. COST BENEFITS:

6.2.1. HARDWARE COST:

No hardware is required as the work can be done on Google Collab, acloud-based platform that provides free access to computing resource

6.3.2 SOFTWARE COST:

- Google Collab is free to use.
- Docker is a free and open-source platform for containerizing software applications.

6.3.3 STAFFING COST:

This task can be done by someone with knowledge of machinelearning and natural language processing.

6.3.4 TOTAL COST:

The total costs of the proposed work are **zero**.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

16.1. CONCLUSION:

16.2. FUTURE ENHANCEMENT:

CHAPTER 8

REFERENC

E

1.