Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8343652

HARMONIZING TEXT AND AUDIO: AI-POWERED EMOTION RECOGNITION AND SENTIMENT ANALYSIS

MS. VIJAYALAXMI KONDAL
Research Scholar, Information Technology, Vidyalankar Institute of Technology, Mumbai, India.

Dr. VIDYA CHITRE
Professor, Information Technology, Vidyalankar Institute of Technology, Mumbai, India

Abstract

In the realm of human-computer interaction, emotion recognition and sentiment analysis play a pivotal role in understanding user experiences and enhancing communication between individuals and machines. This research paper delves into the integration of text and audio-based approaches for emotion recognition and sentiment analysis using advanced Artificial Intelligence (AI) techniques. By harmonizing these modalities, we aim to develop a more comprehensive and accurate understanding of users' emotional states and sentiments. The proposed methodology leverages Natural Language Processing (NLP) techniques for processing textual data and audio signal processing methods for analyzing audio inputs. Our AI-driven framework employs machine learning algorithms, including deep learning models, to capture nuanced emotions and sentiments expressed in both text and audio content. The synergy between these modalities promises to enrich the accuracy and reliability of emotion recognition and sentiment analysis systems. Through experiments and evaluations on diverse datasets, we showcase the effectiveness of the hybrid approach in capturing complex emotional cues. The results demonstrate enhanced accuracy and cross-modal validation, highlighting the potential for real-world applications in fields such as customer sentiment analysis, virtual assistants, and affective computing. This article introduces a hybrid model that combines machine learning and deep learning methodologies for the purpose of emotion identification in text. The model leverages Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) as its deep learning components, while also incorporating Support Vector Machines (SVM) as a machine learning technique. To assess the effectiveness of this approach, the model's performance is gauged across three distinct types of datasets: sentences, tweets, and dialogs. Notably, the proposed hybrid model achieves an impressive accuracy rate of 85.11%.

Keywords: Emotion Recognition, Sentiment Analysis, Artificial Intelligence, Text Analysis, Audio Analysis, Natural Language Processing, Deep Learning, Human-Computer Interaction, Affective Computing, Cross-Modal Analysis.

1) INTRODUCTION

Within the domain of artificial intelligence (AI), there has been a notable focus on the understanding and analysis of human emotions, which has garnered significant attention and promise. The exponential growth of data in contemporary times has paved the way for the development of artificial intelligence (AI) systems that possess the ability to accurately interpret the intricate nuances of human voice and writing. This technological advancement holds immense potential for various industries, including marketing and healthcare, as it may significantly influence their operations and outcomes. The current research aims to explore the effective utilization of artificial intelligence (AI) in recognizing emotions from textual and auditory data, drawing upon the multidisciplinary convergence of linguistics, acoustic studies, and computer methodologies.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8343652

The endeavor to decipher human emotion through communication has a long history, spanning several centuries, and has been informed by various academic fields such as psychology and linguistics [1]. In the present era, the field of machine learning and natural language processing (NLP) has witnessed significant progress, providing an unprecedented potential for expanding and automating the process of decoding. The advancements made in deep learning, specifically in the field of recurrent neural networks (RNN) and transformers, have demonstrated impressive levels of precision when applied to text sentiment analysis [2]. In recent years, there have been significant advancements in the field of audio signal processing and the development of convolutional neural networks (CNN). These advancements have enabled the extraction of complex emotional indicators from spoken language [3].

Nevertheless, despite the considerable advancements gained in text and audio emotion identification systems separately, there is still a lack of exploration in developing an integrated strategy that effectively combines both modalities. The potential for enhanced understanding may arise from the collaborative interaction between various elements of human communication, as emotions are frequently conveyed through an intricate combination of linguistic components, vocal intonation, pitch modulation, and rhythmic patterns.

Ekman [1] identified six primary emotions: joy, sadness, fear, surprise, anger, and disgust. Additionally, emotions can manifest in varied forms such as love and optimism, as depicted in Figure 1 [2]. While human emotions are typically conveyed through facial expressions, gestures, speech, and written text, the latter often lacks the inherent expressiveness of the former modalities. Written text, devoid of tonal and facial cues, poses challenges due to its potential for ambiguity and multifaceted meanings, making emotion recognition particularly intricate.

In recent times, researchers have been examining various approaches to detect emotions in text. These approaches include keyword-based, lexical affinity, learning-based, and hybrid models [3]. In the beginning, solutions based on rules were prevalent, which included approaches such as lexical affinity and keyword-driven algorithms. Over the course of time, there has been a development of learning-based models that have demonstrated improved levels of precision. These models employ diverse methodologies to determine emotions. In order to achieve the highest level of precision, researchers have combined various methodologies, resulting in the development of hybrid models. It is worth noting that deep learning models have greater performance compared to typical machine learning models when applied to large text datasets, whereas machine learning demonstrates superiority in handling smaller datasets. However, it is important to note that there is currently no singular

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8343652

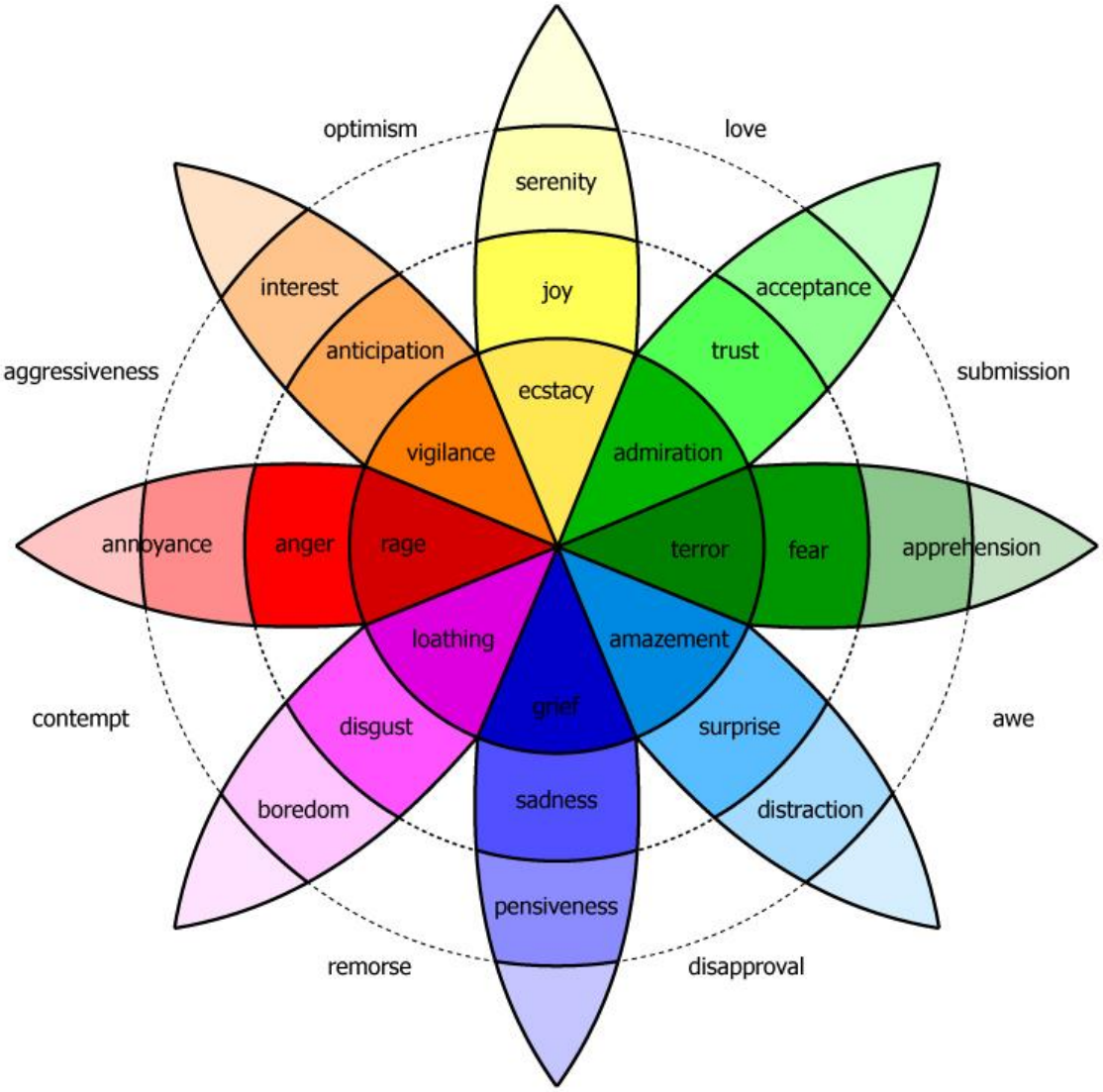

Figure 1: Various Types of Emotions

Method that has been able to accurately decode emotions expressed in written language without any shortcomings.

The current methodologies exhibit various limitations. These include a lack of a comprehensive emotional lexicon, limited word vocabularies, insufficient consideration of semantics-based context, and inadequate extraction of contextual data from sentences, difficulties in detecting certain emotions, slow computation speeds, disregarded feature relations, insufficient data, and high rates of misclassification. Certain models encounter challenges when it comes to handling commonly used emojis, extracting semantic information, and constructing sentences. Although other models have attempted to tackle

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8343652

these challenges, none of them have successfully resolved all of them. The solution we suggest, however, effectively tackles numerous prevailing difficulties.

The detection of emotions plays a crucial role in improving interactions between humans and machines, enabling robots to approximate the understanding of emotions similar to humans. Our proposed model is capable of accurately identifying and interpreting emotions conveyed through textual content, despite the absence of tone or emotional cues often seen in spoken or non-verbal communication. In contrast to most scholars who concentrate on a single dataset, our study incorporates three distinct datasets comprising of plain texts, tweets, and dialogues. The integration of this adaptable text emotion recognition model into other systems is feasible. From a company standpoint, this provides valuable insights into customer feelings expressed in reviews, improves the quality of services offered, enhances the safety of social media users, and delivers additional benefits.

2) LITERATURE SURVEY

Text Base Sentimental Analysis

Seal et al. [4] employed a keyword-based strategy to detect emotions with a primary focus on phrasal verbs using the ISEAR dataset [5]. Their findings indicated discrepancies in the association of phrasal verbs with emotional terms, prompting them to curate their own database. This endeavor led to a remarkable 65% accuracy, though some challenges, such as inadequate emotion keywords and disregard for word semantics, persisted.

Alotaibi [5], on the other hand, adopted a learning-based approach using classifiers such as Logistic Regression, K-Nearest Neighbour (KNN), XG-Boost, and Support Vector Machine (SVM) on the ISEAR database. The author found Logistic Regression to outperform the other classifiers and hinted at the potential of deep learning for model enhancement.

P. Xu et al. [6] introduced Emo2Vec, an innovative method that encodes emotional semantics in vector form. This method was trained within a multitask learning framework on datasets like ISEAR, WASSA, and Olympic. Their results surpassed benchmarks like Convolution Neural Network (CNN) and DeepMoji embedding. Integration of Emo2Vec with Logistic Regression and GloVe showcased even more promising outcomes.

Ragheb et al. [7] proposed a system to detect emotions from textual conversations, relying on Paul Ekman's six emotion classification. Their two-phase methodology consisted of encoding and classification. Utilizing Bi-LSTM units and a self-attention mechanism, the model achieved an impressive F1 score of 75.82%.

M. Suhasini and B. Srinivasu [8] employed machine learning classifiers, namely KNN and Naive Bayes (NB), on the Sentiment 140 corpus for emotion detection. They reported NB's superiority with an accuracy of 72.06% compared to KNN's 55.50%.

M. Hasan et al. [9] combined supervised machine learning and an emotion dictionary for emotion recognition in text. The model encompassed both offline and online strategies and yielded a commendable accuracy of 90%.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8343652

Rodriguez et al. [10] delved into emotion analysis to pinpoint hate speech on social media platforms, aiming to comprehend the underlying emotions in hostile comments.

Cao et al. [11] integrated machine and deep learning methodologies to discern emotions in text, emphasizing the inherent challenges of the task.

Acheampong et al.[12] provided a comprehensive overview of emotion detection (ED) in texts, detailing prevailing techniques.

P. Nandwani and R. Verma [13] focused on the creation of an enriched emotion lexicon to advance the emotion analysis process.

K. Sailunaz and R. Alhajj [14], utilizing Twitter data, explored the realms of sentiment and emotion detection. Their approach harnessed sentiment and emotion scores to offer both generalized and personalized user recommendations based on Twitter engagement.

Speech Base Sentimental Analysis

Several studies have ventured into the realm of emotion recognition from auditory data. The crux of these investigations revolves around the extraction of defining characteristics from a corpus of emotional speech, followed by emotion classification based on these extracted features. The efficacy of emotion classification is intricately tied to the adept extraction of these features. Among the myriad of characteristics explored, spectral characteristics, prosodic characteristics, and their combinations (like the amalgamation of the MFCC acoustic feature with energy prosodic features) have garnered significant attention [15].

Noroozi et al [16]. put forth a comprehensive emotion recognition system pivoting on both visual and auditory signals analysis. Their feature extraction phase employed 88 features, notably the Mel Frequency Cepstral Coefficients (MFCC) and filter bank energies (FBEs). To optimize and pare down the dimensionality of the previously extracted features, they incorporated Principal Component Analysis (PCA).

In another study, Bandela et al. [17] melded the acoustic feature, namely MFCC, with the Teager Energy Operator (TEO) as a prosodic characteristic. This fusion was leveraged to discern five distinct emotions using the GMM classifier and was rooted in the Berlin Emotional Speech database.

Zamil et al.[18] charted a similar trajectory by harnessing spectral characteristics, specifically 13 MFCCs extracted from audio data. Their proposed system set out to classify seven emotions utilizing the Logistic Model Tree (LMT) algorithm. This method culminated in an accuracy rate hovering around 70%.

3) DATASETS

For the textual data, we employed the Stream-of-consciousness dataset, which originated from a study conducted by Pennebaker and King in 1999[19]. This dataset comprises 2,468 unique daily writing entries submitted by 34 psychology students, including 29 females and 5 males. The age of these students varied from 18 to 67, with an average age of 26.4 years. These writing entries were part of an ungraded assignment for a

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8343652

course. Each student had the task of dedicating at least 20 minutes daily to write on a designated topic. The data collection spanned a 2-week summer course from 1993 to 1996, with each participant writing consistently for 10 days. To evaluate the students' personalities, they responded to the Big Five Inventory (BFI), a tool devised by John et al. in 1991. The BFI is a questionnaire containing 44 items, each described by brief statements that measures the five primary personality traits. Respondents rate each item on a scale from 1 (strongly disagree) to 5 (strongly agree). Every data entry in this dataset includes an ID, the respective essay, and classification labels for the Big Five personality traits. Initially, these labels were denoted as 'y' or 'n', representing a high or low score in a specific trait, respectively.

Regarding audio datasets, we utilized the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)[20]. Housing a total of 7,356 files, this substantial dataset, amounting to 24.8 GB, encompasses recordings from 24 professional actors (12 female, 12 male). These actors vocalized two statements with identical lexical content in a neutral North American accent. For speech, the expressions spanned various emotions like calmness, happiness, sadness, anger, fear, surprise, and disgust. In contrast, the song recordings captured emotions such as calm, happy, sad, angry, and fear. Every emotion was depicted at two intensity levels: normal and strong, complemented by a neutral tone. The available modalities for these recordings are Audio-only (16bit, 48kHz .wav), Audio-Video (720p H.264, AAC 48kHz, .mp4), and Video-only (silent). More details can be found [here] (https://zenodo.org/record/1188976#.XCx-tc9KhQI).

4) PROPOSED SCHEME

To develop a comprehensive text-based personality recognition system that identifies the five main personality traits (as per the Big Five Inventory) from textual input.

1. Data Collection and Pre-processing

- Dataset: Utilize the Stream-of-consciousness dataset mentioned earlier.
- Data Augmentation: Incorporate other textual data if needed, ensuring a diverse and balanced dataset for better generalization.

2. Custom Natural Language Preprocessing

- Tokenization: Break the document into individual words or tokens.
- Text Cleaning: Use regular expressions to clean and standardize formulations, ensuring only meaningful content remains. Delete punctuation, which often doesn't contribute to understanding the personality.Convert all tokens to lowercase to ensure uniformity.
- Stopword Removal: Use an enhanced list of stopwords that removes words that are not just generic English stopwords but also other irrelevant words in the context of personality prediction.
- Part-of-Speech Tagging: Apply part-of-speech tags to tokens. This helps in understanding the syntactical importance of each token.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8343652

- Lemmatization: Use part-of-speech tags to accurately lemmatize tokens, ensuring the base or dictionary form of a word is used.
- Sequence Padding:Ensure all input vectors are of the same shape by padding shorter sequences.

3. Feature Engineering and Representation

- Word Embeddings: Utilize 300-dimensional Word2Vec embeddings. These embeddings can either be pre-trained (like Google's Word2Vec or GloVe) or can be trained on the dataset.Consider exploring other advanced embeddings like BERT or ELMo for richer representation.

4. Model Design and Training

- Neural Network Architecture: Use layers suitable for sequential data like LSTM or GRU, given the sequential nature of text. Incorporate attention mechanisms to weigh different parts of the text differently, giving importance to parts that are more indicative of personality traits. Implement dropout for better generalization and to prevent overfitting.
- Training Strategy: Use a split of the dataset for validation to fine-tune hyperparameters. Utilize techniques like early stopping and learning rate decay to enhance training.
- Evaluation Metrics:Apart from accuracy, consider F1-score, recall, and precision for each personality trait, especially if the dataset is imbalanced.

5. Model Evaluation and Testing

Test the model's performance on unseen data. Utilize confusion matrices to understand misclassifications among different personality traits.

6. Feedback Loop and Continuous Learning

Consider implementing a feedback loop where users can validate the model's prediction, providing valuable data for continuous improvement. Regularly retrain the model with new data, ensuring it stays updated with evolving language trends and usage.

7. Deployment

Once satisfied with performance, deploy the model as a web service or API, enabling integration with various applications, such as chatbots, customer service portals, or HR recruitment tools.

Pseudocode for the Described Pipeline

Text-based personality recognition pipeline

Function PersonalityRecognitionPipeline(text_document):
  # 1. Text data retrieving
  raw_data = RetrieveTextData(text_document)
 	
  # 2. Custom natural language preprocessing
  tokens = TokenizeDocument(raw_data)

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8343652

```
    # Cleaning and standardization
    clean_tokens = UseRegularExpressionsForCleaning(tokens)
    # Deletion of punctuation
    tokens_no_punctuation = RemovePunctuation(clean_tokens)
    # Lowercasing the tokens
    lowercase_tokens = ConvertToLowerCase(tokens_no_punctuation)
    # Removal of predefined stopwords
    tokens_without_stopwords = RemoveStopwords(lowercase_tokens)
    # Application of part-of-speech tags on the remaining tokens
    pos_tags = ApplyPOSTags(tokens_without_stopwords)
    # Lemmatization of tokens using part-of-speech tags for more accuracy
    lemmatized_tokens = LemmatizeUsingPOSTags(tokens_without_stopwords,
    pos_tags)
    # Padding the sequences of tokens to fix the shape of the input vectors
    padded_sequence = PadTokenSequences(lemmatized_tokens)

    # 3. Convert padded sequence to 300-dimensional Word2Vec trainable
    embedding
    embedding_vector = ConvertTo300DWord2VecEmbedding(padded_sequence)

    # 4. Prediction using our pre-trained model
    prediction = PreTrainedModelPredict(embedding_vector)
    Return prediction
End Function
```

To develop a state-of-the-art speech emotion recognition system capable of distinguishing between various emotional states from raw audio input.

1. Data Collection and Pre-Processing

- Dataset: Use the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) and consider augmenting with other datasets to ensure diversity and breadth in the training data.
- Signal Enhancement: Implement noise reduction techniques to enhance the clarity of the voice samples.

2. Audio Signal Pre-Processing

- Voice Recording: Ensure recordings are made under controlled environments to minimize external disturbances.
- Audio Signal Discretization: Convert continuous audio signals into discrete signals for further processing.
- Feature Extraction: Log-mel-spectrogram extraction: Extract log-mel-spectrograms from the discrete signals, which are representations of the short-time power spectra of a sound. Consider additional features like MFCCs (Mel-frequency cepstral coefficients), Chroma, and Pitch.
- Segmentation: Split spectrograms using a rolling window technique to get fixed-size input for the model.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8343652

3. Feature Engineering and Representation

- Standardization: Normalize the extracted features so that they have a zero mean and unit variance.
- Time Series Data Representation: Ensure that the data is represented in a manner suitable for time series classification, keeping sequential information intact.

4. Model Design and Training

- Neural Network Architecture: Use Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), especially LSTM or GRU layers, as they have shown promise in speech emotion recognition tasks. Explore the potential of attention mechanisms for weighing different parts of the audio differently. Implement dropout and batch normalization to ensure robust training and prevent overfitting.
- Training Strategy: Divide the dataset into training, validation, and testing sets. Use early stopping and adaptive learning rates for efficient training.
- Evaluation Metrics: Use accuracy, F1-score, precision, and recall to evaluate model performance. Confusion matrices can also help understand the model's predictions across different emotions.

5. Model Evaluation and Testing

Use a hold-out test set to evaluate the model's performance. Implement techniques like k-fold cross-validation for robust evaluation.

6. Feedback Loop and Continuous Learning

Deploy the model in real-world applications and gather feedback from users. Regularly retrain the model with new data samples and user feedback.

7. Deployment

Deploy the trained model as an API or a module that can be integrated into applications like voice assistants, customer support bots, or therapy chatbots to understand the emotional state of the user.

Speech Emotion Recognition Pipeline

```
Function SpeechEmotionRecognitionPipeline(voice_data):
    # 1. Voice recording
    recorded_voice = RecordVoice(voice_data)

    # 2. Audio signal discretization
    discrete_signal = DiscretizeAudioSignal(recorded_voice)

    # 3. Log-mel-spectrogram extraction
    log_mel_spectrogram = ExtractLogMelSpectrogram(discrete_signal)

    # 4. Split spectrogram using a rolling window
    split_spectrograms = SplitUsingRollingWindow(log_mel_spectrogram)
```

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8343652

```
    # 5. Make a prediction using our pre-trained model
    predictions = []
    For each spectrogram in split_spectrograms:
        prediction = PreTrainedModelPredict(spectrogram)
        Append prediction to predictions

    Return predictions
End Function
```

5) DEEP AND MACHINE LEARNING ALGORITHM USED

A. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs)[21], a class of deep learning algorithms, are specifically designed for processing structured grid data, such as an image. They are primarily known for their applications in image recognition, but have been utilized in various other domains like natural language processing, video analysis, and even bioinformatics.

Key Components of CNNs

Input Layer

The initial layer that accepts the raw data (e.g., pixel values of an image).

Convolutional Layer

This is the core building block of a CNN. It involves a filter (also called a kernel) that moves across the input data (such as an image) to produce a feature map or convolved feature. The objective of this layer is to identify various features in the input, such as edges or textures.

Activation Function

After convolution, an activation function is applied, like the Rectified Linear Unit (ReLU). It introduces non-linearity to the model, enabling it to learn from the error and better generalize across unseen data.

Pooling Layer

This layer reduces the spatial dimensions of the convolved feature, which reduces the computational complexity for upcoming layers.

Max pooling and average pooling are the most commonly used techniques.

Fully Connected Layer

Neurons in a fully connected layer have connections to all activations in the previous layer. It's typically used as the final layer in a CNN, producing the desired output dimensions (like class scores in classification tasks).

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8343652

Output Layer

Provides the final prediction. In the case of classification, it would often utilize the softmax activation function to output a probability distribution over classes.

Training

CNNs are trained using backpropagation, similar to other neural networks. The objective during training is to adjust the model's weights to minimize the difference between the predicted output and the actual target values, typically using a gradient descent optimization algorithm.

Advantages of CNNs

- Parameter Sharing: A feature detector (filter) that's useful in one part of the image can also be useful for other parts.
- Sparsity of Connections: In each layer, each output value depends only on a small number of input values, making the computation efficient.
- Translation Invariance: Due to the nature of convolution, a learned feature can be recognized anywhere in the image.

B. Long Short-Term Memory (LSTM) Networks

LSTM (Long Short-Term Memory)[22][23] is a type of recurrent neural network (RNN) architecture. RNNs are designed to recognize patterns in sequences of data, such as time series or natural language. However, traditional RNNs have difficulty in learning long-range dependencies in the data. LSTMs were designed to overcome this limitation and are particularly effective at learning from long-term sequences, which are typically challenging for other neural networks.

Key Components of LSTMs

Cell State

It's the "memory" of the LSTM and runs straight down the entire chain, with only a few linear interactions. It carries information throughout the processing of the sequence.

Forget Gate

Decides what information from the cell state should be thrown away or kept. It uses a sigmoid activation function, which outputs values between 0 (forget) and 1 (keep).

Input Gate

- The input gate updates the cell state with new information. It contains two parts:
- A sigmoid layer which decides which values to update.
- A tanh layer which creates a vector of new candidate values.

Output Gate

Based on the cell state and the input, it decides what the next hidden state should be.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8343652

The hidden state then can be used for predictions, and it is also passed to the next LSTM cell.

Hidden State

It carries information similar to the cell state but is exposed to the network, unlike the cell state which is used internally by the LSTM cell.

Advantages of LSTMs

- Learning Long-Term Dependencies: They are specifically designed to combat the vanishing gradient problem in traditional RNNs, allowing them to learn from long sequences.
- Memory Cell: The cell state, or the memory cell, can maintain information in memory for long periods of time.
- Gating Mechanism: The input, forget, and output gates in LSTMs enable the model to regulate the flow of information.

C. Support Vector Machines (SVM)

Support Vector Machines (SVM) [23][24] are a type of supervised machine learning algorithm that can be used for classification or regression problems. It performs classification by finding the hyperplane that best divides a dataset into classes.

Key Concepts

Hyperplane

In an N-dimensional space (where N is the number of features), a hyperplane is a flat affine subspace of dimension N−1. In SVM, the objective is to find the optimal hyperplane that separates different classes.

Support Vectors

These are the data points that lie closest to the hyperplane and influence its orientation and position. The distance between the hyperplane and the support vectors is maximized. Removing a support vector may affect the position of the hyperplane, but other data points will not.

Margin

It is the distance between the hyperplane and the nearest data point from either class. The goal of an SVM is to maximize this margin, ensuring the greatest possible separation between the hyperplane and any data point.

Kernel Trick

In cases where data isn't linearly separable in its current dimension, the data is transformed into a higher dimension where it becomes linearly separable. This transformation is achieved using a kernel function. Popular kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8343652

Soft Margin vs. Hard Margin

Hard Margin: Assumes that the data is perfectly separable, which is often not the case in real-world scenarios.

Soft Margin: Allows some misclassifications, aiming to find a balance between margin maximization and loss.

Advantages of SVM

- Effective in High Dimensional Spaces: Works well when the number of features is greater than the number of samples.
- Uses a Subset of Training Points: Only the support vectors are used to specify the hyperplane, making it memory efficient.
- Versatility: Different kernel functions can be specified for the decision function.

6) RESULT AND DISCUSSION

We have conducted a series of experiments employing diverse methodologies in order to achieve optimal accuracy for our novel model. Our investigations encompass emotion classification using a machine learning paradigm, a deep learning paradigm, and a hybrid model approach. These experiments were carried out on a multitext dataset comprising sentences, tweets, and dialogs. In our study, we aimed to explore various approaches for emotion classification on a multitext dataset that includes sentences, tweets, and dialogs. We utilized a pipeline that transformed input text into vectors, which were then employed to train different classifiers, including Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and Long Short-Term Memory networks (LSTM). The results obtained from our experiments are presented and discussed below.

Text-based Emotion Classification

For the text-based approach, we first processed the input text using the pipeline to convert it into numerical vectors. These vectors were then utilized to train SVM and LSTM classifiers. The performance metrics for these classifiers are summarized in the table below:

Classifier	Precision	Recall	F1 score	Accuracy
SVM	81.45	78.36	79.67	78.97
LSTM	77.12	71.25	81.17	85.02

The results indicate that the SVM classifier achieved a commendable precision of 81.45% along with a recall of 78.36%, leading to an F1 score of 79.67% and an overall accuracy of 78.97%. On the other hand, the LSTM classifier exhibited a precision of 77.12% and a recall of 71.25%, resulting in an F1 score of 81.17%. The LSTM classifier displayed a higher accuracy of 85.02%, showcasing its effectiveness in capturing sequential dependencies in the text data.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8343652

Audio-based Emotion Classification

In the audio-based scenario, we followed a similar pipeline, converting audio data into numerical vectors. These vectors were employed to train SVM, CNN, and LSTM classifiers. The performance metrics for this approach are summarized in the table below:

Classifier	Precision	Recall	F1 score	Accuracy
SVM	61.45	68.60	67.03	68.03
LSTM	73.12	74.29	83.32	90.36

The results for the audio-based approach indicate that the SVM classifier achieved a precision of 61.45%, a recall of 68.60%, and an F1 score of 67.03%. The overall accuracy for the SVM classifier was 68.03%. The LSTM classifier, however, demonstrated improved performance, with a precision of 73.12% and a recall of 74.29%, yielding a high F1 score of 83.32% and an impressive accuracy of 90.36%. This suggests that the LSTM model effectively captures temporal patterns in audio data, leading to superior classification results.

Comparing the two approaches, it is evident that the LSTM classifier consistently outperformed the SVM classifier in both the text-based and audio-based classification tasks. The LSTM's ability to capture sequential patterns and dependencies in the data proved beneficial for accurately classifying emotions. Additionally, the audio-based approach demonstrated higher accuracy compared to the text-based approach, showcasing the potential of audio data in emotion classification.

7) CONCLUSION

In this paper, proposed a text-based emotion recognition model. The proposed model is a combination of deep learning and machine learning approaches. Comparing the two approaches, it is evident that the LSTM classifier consistently outperformed the SVM classifier in both the text-based and audio-based classification tasks. The LSTM's ability to capture sequential patterns and dependencies in the data proved beneficial for accurately classifying emotions. Additionally, the audio-based approach demonstrated higher accuracy compared to the text-based approach, showcasing the potential of audio data in emotion classification. Furthermore, the hybrid model approach mentioned earlier, which involves combining different classifiers, could potentially lead to even better results. It might leverage the strengths of each classifier and mitigate their weaknesses, resulting in enhanced overall performance. In conclusion, our experiments underscore the significance of model selection and data representation in emotion classification tasks. The LSTM classifier emerged as a strong candidate, especially for audio-based data, demonstrating its effectiveness in capturing complex patterns. These findings provide valuable insights for the design and implementation of emotion classification systems using different types of data.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8343652

Reference

1) P. Ekman, “Basic emotions,” Handbook of cognition and emotion, vol. 98, no. 45-60, p. 16, 1999.

2) R. Plutchik, “The nature of emotions,” American Scientist, vol. 89, no. 4, p. 344, 2001.

3) C. R. Chopade, “Text based emotion recognition: a survey,” International Journal of Science and Research, vol. 2, no. 6, pp. 409–414, 2015.

4) D. Seal, U. K. Roy, and R. Basak, “Sentence-level emotion detection from text based on semantic rules,” Information and Communication Technology for Sustainable Development, Springer, Singapore, pp. 423–430, 2020.

5) S. M. Mohammad and F. Bravo-Marquez, “WASSA-2017 Shared Task on Emotion Intensity,” 2017, http://arxiv.org/abs/1708.03700.

6) P. Xu, A. Madotto, C. S. Wu, J. H. Park, and P. Fung, “Emo2vec: learning generalized emotion representation by multi-task training,” 2018, http://arxiv.org/abs/1809.04505.

7) W. Ragheb, J. Azé, S. Bringay, and M. Servajean, “Attention-based modeling for emotion detection and classification in textual conversations,” 2019, http://arxiv.org/abs/1906.07020.

8) M. Suhasini and B. Srinivasu, “Emotion detection framework for twitter data using supervised classifiers,” Data Engineering and Communication Technology, Springer, Singapore, pp. 565–576, 2020.

9) M. Hasan, E. Rundensteiner, and E. Agu, “Automatic emotion detection in text streams by analyzing twitter data,” International Journal of Data Science and Analytics, vol. 7, no. 1, pp. 35–51, 2019.

10) A. Rodriguez, Y. L. Chen, and C. Argueta, “FADOHS: framework for detection and integration of unstructured data of hate speech on facebook using sentiment and emotion analysis,” IEEE Access, vol. 10, pp. 22400–22419, 2022.

11) L. Cao, S. Peng, P. Yin, Y. Zhou, A. Yang, and X. Li, “A survey of emotion analysis in text based on deep learning,” in Proceedings of the 2020 IEEE 8th International Conference on Smart City and Informatization (iSCI), pp. 81–88, IEEE, Guangzhou, China, December 2020.

12) F. A. Acheampong, C. Wenyu, and H. Nunoo-Mensah, “Textbased emotion detection: a,” Engineering Reports, vol. 2, no. 7, Article ID e12189, 2020.

13) A. S. Navarrete, C. Martinez-Araneda, C. Vidal-Castro, and C. Rubio-Manzano, “A novel approach to the creation of a labelling lexicon for improving emotion analysis in text,” The Electronic Library, vol. 39, 2021.

14) K. Sailunaz and R. Alhajj, “Emotion and sentiment analysis from Twitter text,” Journal of Computational Science, vol. 36, Article ID 101003, 2019.

15) P. Sharma, V. Abrol, A. Sachdev, and A. D. Dileep, "Speech emotion recognition using kernel sparse representation based classifier," in 2016 24th European Signal Processing Conference (EUSIPCO), pp. 374-377, 2016.

16) F. Noroozi, M. Marjanovic, A. Njegus, S. Escalera, & G. Anbarjafari, "Audio-visual emotion recognition in video clips," IEEE Transactions on Affective Computing, 2017.

17) S. R. Bandela and T. K. Kumar, "Stressed speech emotion recognition using feature fusion of teager energy operator and MFCC," in 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-5, 2017.

18) Zamil, Adib Ashfaq A., et al. "Emotion Detection from Speech Signals using Voting Mechanism on Classified Frames." 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST). IEEE, 2019.

ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 09-2023
DOI: 10.5281/zenodo.8343652

19) J. W. Pennebaker and L. A. King, "Linguistic styles: Language use as an individual difference," Journal of Personality and Social Psychology, vol. 77, no. 6, pp. 1296–1312, 1999.

20) S. R. Livingstone and F. A. Russo, "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)," 2018. [Online]. Available: https://zenodo.org/record/1188976#.XCx-tc9KhQI.

21) LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4), 541-551.

22) The LSTM was introduced by Sepp Hochreiter and Jürgen Schmidhuber in their seminal 1997 paper:

23) Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780.

24) Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning theory (pp. 144-152).

25) Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Series in Statistics. Springer New York Inc.