EMOTION DETECTION: A COMPARATIVE ANALYSIS OF LSTM, BERT, AND CNN MODELS

Charan Sumanth Pulleti
MS in Data Science
University of New Haven
cpull2@unh.newhaven.edu

Abstract - Emotion detection, also known as sentiment analysis, plays a pivotal role in understanding human expressions conveyed through text data. This project aims to develop robust machine learning models for accurately detecting and classifying emotions in textual content. Three distinct models, including Long Short-Term Memory (LSTM), Bidirectional Encoder Representations from Transformers (BERT), and Convolutional Neural Network (CNN), were employed to explore diverse approaches to emotion detection. Each model architecture was meticulously designed to leverage the strengths of recurrent neural networks, transformer-based models, and convolutional neural networks in processing textual data and extracting meaningful features. Real-world applications of emotion detection across domains such as customer service, education, mental health monitoring, market research, and human-computer interaction underscore the significance and practical implications of this project.

Through the integration of deep learning techniques and transformer-based architectures, this research contributes to advancing the field of emotion detection and its applications in understanding human behaviour through textual communication.

1. INTRODUCTION

In the era of digital communication, understanding the intricacies of human emotions expressed through text has become increasingly vital across various domains. Emotion detection, also referred to as sentiment analysis, offers a powerful tool for unravelling the emotional content embedded within textual data. This project delves into the realm of robust models capable of accurately detecting and classifying emotions in text.

The primary objective of this endeavour is to enhance our comprehension of human emotions as conveyed through written communication. By leveraging advanced techniques, including Long Short-Term Memory (LSTM), Bidirectional Encoder Representations from Transformers (BERT), and Convolutional Neural Network (CNN) architectures, we aim to create models proficient in discerning a wide spectrum of emotions.

This introduction sets the stage for exploring the multifaceted landscape of emotion detection, highlighting its relevance across diverse domains such as customer feedback analysis, healthcare, market research, and sentiment analysis. By embarking on this journey, we endeavour to unravel the complexities of human emotions encoded within textual data, paving the way for applications that offer deeper insights into human behaviour and interaction.

2. PROPOSED IDEA

This project aims to develop and compare multiple models, including LSTM, BERT, and CNN, for the task of emotion detection in textual data. The primary objective of this project is to develop a model capable of accurately detecting and classifying emotions expressed in text data. By achieving high accuracy in emotion detection tasks, we seek to

enhance understanding and analysis of human emotions as expressed through written communication.

Leveraging a carefully curated dataset annotated with six emotion classes, the project will explore various preprocessing techniques and model architectures to achieve high accuracy in classifying emotions such as sadness, joy, love, anger, fear, and surprise.

The ultimate goal is to enhance understanding and analysis of human emotions expressed through written communication, with potential applications spanning customer feedback analysis, healthcare, market research, and sentiment analysis.

a. Dataset overview

The dataset utilized in this project comprises textual data annotated with corresponding emotion labels, facilitating the task of emotion detection. With a total of 16,000 samples distributed across six distinct emotion classes—sadness, joy, love, anger, fear, and surprise—the dataset provides a diverse and comprehensive foundation for model training and evaluation. To ensure data integrity and consistency, preprocessing steps were applied, including the removal of duplicate entries.

The dataset is partitioned into three subsets: training, validation, and test, each serving a specific role in the model development pipeline. Through meticulous curation and annotation, the dataset serves as the cornerstone for training and evaluating machine learning models capable of accurately detecting and classifying emotions expressed in textual content.

b. Data preprocessing

1. Duplicate Entry Removal - Removing duplicate entries from the training data is a crucial step in data preprocessing to ensure the integrity and effectiveness of machine learning models. Duplicate instances within the training dataset can introduce bias and skew the model's learning process, potentially leading to overfitting and inaccurate predictions. By identifying and eliminating duplicate entries, data integrity is preserved, and the model's ability to generalize to unseen data is enhanced. This process promotes a more robust and reliable training phase, ultimately improving the model's performance and accuracy in emotion detection tasks.

2. Label Mapping - Mapping numerical labels to emotion descriptions are a data preprocessing step aimed at enhancing the interpretability of the dataset. This step facilitates better understanding and analysis of the data by converting numerical labels, which are often used for computational purposes, into corresponding emotion descriptions. By mapping numerical labels to meaningful emotion descriptions, the data becomes more human-readable, allowing researchers and practitioners to interpret the emotions represented in the dataset more intuitively.

3. Duplicate Entry Validation - Duplicate entry checking in the test and validation datasets is essential to uphold the integrity of the evaluation process in machine learning tasks. Duplicates in these datasets can lead to skewed performance metrics and inaccurate assessments of model generalization. By ensuring that each instance in the test and validation sets is unique, researchers and practitioners can confidently evaluate the model's performance on unseen data, thereby obtaining reliable insights into its predictive capabilities. This step mitigates the risk of data leakage and ensures that the model's performance metrics accurately reflect its ability to generalize to new, unseen instances, bolstering the credibility of the evaluation process.

4. Data Partitioning - Data partitioning, also known as data splitting, is a crucial step in machine learning model development. It involves dividing the available dataset into multiple subsets for different purposes, typically training, validation, and testing.

- Training Data: This subset is the largest and trains the model by learning patterns and relationships between textual input and emotion labels.

- Validation Data: Used for fine-tuning model hyperparameters and preventing overfitting, this subset evaluates the model's generalization.
- Testing Data: The final evaluation is performed on this independent subset, assessing the model's ability to classify emotions accurately in real-world scenarios.

The data partitioning process ensures that the machine learning model is trained, validated, and tested on distinct datasets, enabling robust evaluation of its performance and generalization capabilities. This step is crucial for building reliable and effective machine learning models for emotion detection tasks.

5. Data Curation and Annotation - Annotation and data curation are essential steps in getting a dataset ready for machine learning applications, especially emotion detection. Carefully choosing and arranging the dataset to guarantee its quality, applicability, and representativeness for the intended purpose is known as data curation.

On the other hand, annotation entails assigning pertinent labels to the data in this case, emotion labels in order to offer the oversight required for machine learning model training. To ensure accuracy and consistency, human annotators may need to label data samples as part of this procedure manually.

Researchers can guarantee that the dataset is well-prepared and provides a dependable basis for training and assessing machine learning models, which will ultimately result in more accurate and efficient outcomes in emotion detection tasks, by devoting time to data curation and annotation.

6. Tokenization - In tokenization, a text is divided into smaller units, usually words or sub words, and each unit is given a distinct numerical identification. Through this procedure, the text data can be encoded as integer sequences that machine learning models can process. By capturing the semantic meaning of words, tokenization aids in the model's comprehension of the text data's structure.

7. Padding - Neural network models need uniformly lengthen inputs for various NLP applications. Nonetheless, variations in phrase or document lengths frequently cause textual data to vary in length. To keep sequences constant in length, padding entails appending special tokens, often zeros, to the start or finish of the sequence. If sequences get longer than you want, you can also truncate them to a fixed length. In order for the neural network model to process input sequences effectively, padding makes sure that all of the sequences have the same size.

c. Model architectures

In this project, three distinct models were employed for emotion detection:

- LSTM (Long Short-Term Memory)
- BERT (Bidirectional Encoder Representations from Transformers)
- CNN (Convolutional Neural Network).

Each model offers unique advantages and architectures suited for processing textual data and extracting meaningful features.

I. Long Short-Term Memory (LSTM)

LSTM is a type of recurrent neural network (RNN) designed to handle sequential data, making it particularly well-suited for processing text. Unlike traditional RNNs, LSTM networks have the ability to retain information over long sequences, thanks to their gating mechanisms.

These gates allow the network to selectively remember or forget information, enabling it to capture long-term dependencies and patterns in the input data. In the context of emotion detection, LSTM models excel at capturing contextual information and temporal dependencies present in textual data, which are essential for accurately classifying emotions expressed over sequences of words.

LSTM architectures typically consist of multiple LSTM cells stacked together, along with additional layers such as embedding layers for converting words into dense numerical vectors. During training, LSTM models learn to extract relevant features from the input text and use them to predict the corresponding emotion labels. Despite their effectiveness, LSTM models may face challenges with processing very long sequences and capturing subtle nuances in text, which can limit their performance in certain scenarios.

II. Bidirectional Encoder Representations from Transformers (BERT)

BERT is a transformer-based model that has revolutionized natural language processing tasks by leveraging bidirectional attention mechanisms. Unlike traditional models that process text in a unidirectional manner, BERT can capture context from both preceding and succeeding words in a sentence, enabling it to generate rich contextual representations of words.

This ability to capture bidirectional context makes BERT highly effective for tasks such as sentiment analysis and emotion detection, where understanding the surrounding context is crucial for accurately interpreting emotions expressed in text.

BERT architecture consists of multiple transformer layers, each comprising self-attention and feedforward neural network sublayers. During training, BERT learns to generate contextual representations of words by predicting missing words in a sentence based on the surrounding context. These pre-trained representations can then be fine-tuned on specific tasks, such as emotion detection, by adding task-specific output layers.

BERT models have demonstrated remarkable performance in various NLP tasks, often surpassing human-level performance on benchmark datasets. However, fine-tuning BERT models can be computationally expensive and may require large amounts of annotated data for optimal performance.

III. Convolutional Neural Network (CNN)

CNNs are a class of deep learning models that have been widely used for image classification tasks but have also shown effectiveness in processing sequential data such as text. In the context of emotion detection, CNN models leverage convolutional filters and pooling layers to capture meaningful patterns and features within the input text. These filters act as feature detectors, identifying important patterns at different levels of abstraction, while pooling layers aggregate the extracted features, reducing the dimensionality of the input data.

CNN architectures for text typically involve stacking multiple convolutional and pooling layers, followed by fully connected layers for classification. During training, CNN models learn to extract hierarchical features from the input text, capturing both local and global patterns that are indicative of different emotions. CNNs are known for their computational efficiency and ability to process input data in parallel, making them well-suited for real-time applications where rapid and accurate emotion detection is paramount.

However, CNNs may struggle with capturing long-range dependencies in text compared to models like LSTM and BERT, which can affect their performance on certain emotion detection tasks.

3. TECHNICAL DETAILS

For LSTM Model

Model Architecture Setup - The LSTM model is constructed using the Keras Sequential API. The architecture includes an embedding layer to convert words into dense numerical vectors, a SpatialDropout1D layer for regularization, an LSTM layer with 64 units for sequence processing, and a Dense output layer with SoftMax activation for multi-class classification.

Model Compilation: The model is compiled using the Adam optimizer with a learning rate of 0.001. "*Sparse_categorical_crossentropy"* is the loss function used, which is suitable for multi-class classification tasks. Additionally, the model's

performance is evaluated based on accuracy during training.

Model Training: The model is trained using the training data (X_train_pad) and corresponding labels (y_train) for 10 epochs with a batch size of 64. Training progress is monitored on the validation data (X_val_pad, y_val) to assess performance and prevent overfitting.

Model Evaluation: After training, the model's performance is evaluated on the test data (X_test_pad, y_test) to assess its generalization ability. The test loss and accuracy are computed and printed to evaluate the model's performance on unseen data.

For BERT Model

Model Definition: The BERT model for sequence classification is initialized using the BertForSequenceClassification class from the Hugging Face Transformers library. This model is pre-trained on a large corpus and fine-tuned for the emotion detection task with the specified number of output labels.

Optimizer Setup: AdamW optimizer is employed to optimize the model's parameters during training. The learning rate and epsilon values are set based on best practices for fine-tuning BERT models.

Training Loop: The model is trained using a DataLoader to efficiently load batches of data. During each epoch, the training data is iterated over, and gradients are computed and updated using backpropagation. The training loss is monitored to assess the model's performance.

Validation: After each epoch, the model's performance is evaluated on the validation dataset to monitor for overfitting. The validation loss is calculated to gauge the model's generalization ability.

Model Evaluation: Once training is complete, the model is evaluated on the test dataset to assess its accuracy. Predicted labels are compared with true labels to compute classification metrics such as precision, recall, and F1-score.

For CNN Model

Model Architecture: The CNN model architecture is defined using the Keras Sequential API. It consists of an embedding layer followed by one or more convolutional layers with max-pooling. These layers are designed to extract features from the input sequences.

Compilation: The model is compiled with appropriate loss function, optimizer, and evaluation metrics using the compile method. In this case, categorical cross-entropy is chosen as the loss function and Adam optimizer is used.

Training: The model is trained on the training data using the fit method. During training, the model learns to map input sequences to their corresponding emotion labels by adjusting its parameters based on the computed loss.

Evaluation: After training, the model is evaluated on the validation and test datasets to assess its performance. The evaluate method is used to compute the loss and accuracy metrics on these datasets.

4. RESULTS

The performance of the emotion classification models was evaluated on the test dataset using various metrics.

a. Classification Report: Metrics like precision, recall, and F1-score for every class in the dataset are included in the classification report, which offers a thorough overview of the model's performance. Recall computes the percentage of true positive predictions among all real positive instances, whereas precision quantifies the percentage of true positive forecasts across all positive predictions. The F1-score offers a fair assessment of the accuracy of the model by taking the harmonic mean of recall and precision. This report is helpful in determining how well the model works for each class and in seeing any disparities or flaws in the way the model predicts outcomes across various emotional categories.

b. Confusion Matrix: This visualization tool shows the number of true positive, true negative, false positive, and false negative predictions for each class, enabling a thorough analysis of the model's performance. It gives a clear picture of the model's performance in differentiating between classes as well as the locations of errors. It is possible to spot misclassification trends and learn which emotion classes the model might have trouble correctly predicting by examining the confusion matrix.

c. Accuracy: An essential parameter for assessing a classification model's overall performance is accuracy. Out of all the cases in the dataset, it calculates the percentage of correctly classified instances. Although accuracy offers a broad indicator of the model's efficacy, it could not always precisely represent the model's performance, particularly when imbalances in classes or disproportionate costs associated with misclassification are present.

For LSTM Model

- Test Accuracy: 91.10%
- F1-score, Precision, and Recall: For the majority of emotion classes, the LSTM model demonstrated good recall and precision scores; it performed especially well at predicting happiness and sorrow. It did, however, perform worse when it came to foreseeing surprise and terror, suggesting room for development.
- Strengths: Demonstrated strong performance in predicting joy and sadness emotions.
- Areas for Improvement: Struggled with emotions like fear and surprise, indicating the need for fine-tuning in those areas.

```
63/63 [==============================]
Test Loss: 0.27731403708457947
Test Accuracy: 0.9110000133514404
```

For BERT Model

- Test Accuracy: 92%
- Precision, Recall, and F1-score: The BERT model demonstrated robust performance across all emotion classes, with accuracy reaching 92%. It exhibited high precision, recall, and F1-score metrics for most classes, indicating its effectiveness in accurately classifying various emotional states in textual data.
- Strengths: Showed robustness in recognizing various emotions, with precision, recall, and F1-score metrics averaging above 0.90 for most classes.
- Areas for Improvement: Some emotions like love and surprise had lower performance compared to others, suggesting potential areas for refinement.

```
from sklearn.metrics import accuracy_score

predicted_labels_bert = val_predicted_labels

accuracy_bert = accuracy_score(y_val, predicted_labels_bert)

print("BERT Model Accuracy:", accuracy_bert)
```

```
BERT Model Accuracy: 0.92
```

For CNN Model

- Test Accuracy: 91.60%
- F1-score, Precision, and Recall: With an accuracy of 91.60%, the CNN model demonstrated remarkable performance in the classification of emotions as well. In most emotion classes, it had great recall and precision scores; it was especially good at identifying happiness and sadness.
- Strengths: Exhibited high precision and recall scores across most emotional classes, particularly excelling in recognizing sadness and joy.
- Areas for Improvement: Slight variations in performance for less frequent emotions like love and surprise were observed.

```
63/63 [==============================]
Test Loss: 0.2865625023841858
Test Accuracy: 0.9160000085830688
```

5. CONCLUSION

In conclusion, the approaches for emotion recognition have produced encouraging findings for a variety of models. We secured high-quality training data by carefully selecting and annotating datasets, providing a strong basis for model development. Classification reports and confusion matrices were used to identify areas for improvement in each model's ability to recognize different emotions. All models demonstrated the potential of algorithmic learning to comprehend and classify intricate human emotions in textual data, notwithstanding their variations. In the future, more improvements and investigations into model architectures and training methods may improve our capacity to identify emotions, leading to a variety of uses in sentiment analysis, mental health monitoring, and other fields.

In comparing the LSTM, BERT, and CNN models for emotion detection, each approach offers unique advantages and challenges. The LSTM model, with its sequential processing capability, demonstrates robust performance, particularly in capturing temporal dependencies in textual data. Its accuracy of 91% on the test dataset underscores its effectiveness in recognizing various emotions. BERT, a transformer-based model, excels in contextual understanding, achieving an impressive accuracy of 92%. Its ability to capture nuanced semantics contributes to its strong performance, especially in distinguishing between subtle emotional cues. On the other hand, the CNN model showcases the power of convolutional neural networks in capturing local patterns, achieving an accuracy of 91% as well. BERT's greater accuracy is probably a result of its skill at efficiently capturing contextual information using language representations currently provided. Compared to LSTM and CNN, this enables it to recognize complexities in text input more effectively, resulting in more accurate emotion classification.

6. FUTURE WORK

Other directions could be explored in the future to improve the emotion-detecting system. First, investigating ensemble techniques such as LSTM, BERT, and CNN that integrate the advantages of several models may enhance overall performance and robustness. Furthermore, optimizing pre-trained language models such as BERT using domain-specific datasets may improve the system's capacity to identify emotions in niche settings like customer service or healthcare. Moreover, including multimodal features like audio or video data inside textual data may offer richer input for emotion recognition and, ultimately, more accurate predictions. Additionally, researching methods for managing unbalanced datasets and uncommon emotions may enhance the system's efficacy and generalization even more.

Ultimately, increasing user happiness and acceptance would be facilitated by using the system in real-world applications and gathering user feedback to iteratively improve the models and interface design. Future approaches for emotion identification and its applications across domains seem promising as long as machine learning and natural language processing continue to progress.

7. REFERENCES

- https://link.springer.com/article/10.1007/s42979-021-00815-1
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10009917/
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9824303/
- https://ieeexplore.ieee.org/document/9679993
- https://researchgate.net/publication/346111006_Overview_of_the_Transformer-based_Models_for_NLP_Tasks
- https://aclanthology.org/2022.findings-emnlp.375.pdf

8. GITHUB LINK

- https://github.com/Charan4767/Emotion-Detection