Deep Learning Approaches for Emotion-Based Sentiment Analysis on Twitter Data

1st Łukasz Chrostowski
Department of Mathematics (DMat)
University of Aveiro (UA)
Aveiro, Portugal
lukasz.chrostowski@ua.pt

2nd Dominika Eder
Department of Mathematics (DMat)
University of Aveiro (UA)
Aveiro, Portugal
dominika.eder@ua.pt

***Abstract*—This study explores the application of Long Short-Term Memory (LSTM) neural networks for emotion recognition in text data, an area gaining increasing relevance in enhancing human-machine interaction. Leveraging a comprehensive dataset of English Twitter messages annotated with fundamental emotions, we developed both non-augmented and augmented LSTM models to classify text-based emotions accurately. Our results demonstrate that the LSTM models, particularly when augmented, perform competitively with, and in some aspects superior to, state-of-the-art models like DistilBERT. This paper not only presents a detailed comparative analysis of LSTM performance against traditional and contemporary models but also discusses the impact of data augmentation techniques on the precision and recall of the classification tasks. These findings underscore the potential of LSTM networks to effectively process and classify complex emotional data in texts, providing substantial contributions to the development of more empathetic and responsive AI systems.**

***Index Terms*—Emotion Recognition, LSTM, Neural Networks, Text Classification, Data Augmentation, Human-Machine Interaction, Sentiment Analysis, Deep Learning.**

I. Introduction

Emotion classification using deep learning is a compelling topic due to its significant impact on enhancing human-machine interaction. The ability to accurately recognize and respond to human emotions can greatly improve the effectiveness of AI systems, such as chatbots and robots, in various applications.

The foundational work of psychologist Paul Eckman in the 1970s, who identified six basic emotions universally experienced across cultures—happiness, sadness, disgust, fear, surprise, and anger—serves as a cornerstone for modern emotion classification efforts. Implementing such knowledge into AI systems enables them to detect and appropriately react to human emotions, thus positively influencing the behavior and mood of individuals they interact with.

Emotion detection is a subset of sentiment analysis, a field that has gained substantial attention due to advancements in machine learning and the availability of large datasets from social networks. Research in sentiment analysis often focuses on analyzing social media content, such as comments on Twitter and Facebook, to understand public sentiment and identify antisocial behavior and toxicity.

Beyond social media, emotion recognition plays a crucial role in mental health, providing insights into patients' emotional well-being and aiding in personalized treatment plans. It also assists policymakers in gauging public sentiment towards political figures and policies, enabling more responsive and effective governance.

A. State of the Art

1) Article nr 1: Detection of emotion by text analysis using machine learning: [1]

In the work (Chatterjee et al., 2019) a deep learning approach called sentiment and semantic LSTM (SS-LSTM) was proposed. They evaluated various deep learning techniques (CNN, LSTM) and various forms of text data representation (Word2Vec, GloVe, FastText, as well as Sentiment Specific Word Embedding). Their approach outperformed most basic machine learning algorithms. The emotions were explored also in Khanpour and Caragea (2018). They proposed a computational model that combines the strengths of CNN, LSTM, and lexical approaches to capture hidden semantics in text messages.

The novel neural network approach was proposed in Kratzwald et al. (2018) as a bi-directional LSTM (BiLSTM) network that can make predictions based on texts of different lengths. Their innovation is two-way text processing, layer extraction as a means of regularization, and a weighted loss function. An extension of transfer learning called sent2affect was also proposed. Their results were comparable to state-of-the-art research using classical machine learning algorithms – the SVM and the random forest of decision trees.

In this article the text data, which is the conversational content of social networks, was labeled with categories of emotions: love, joy, surprise, sadness, anger, and fear. The dataset was pre-processed using Tokenizer API of TensorFlow Keras, and by Padding for transformation of all sentences to the same length which is a condition for neural network generation. They have designed and trained a detection model based on neural networks using a combination of CNN based on 1D convolution - Conv1D and RNN network – LSTM. The model was learned to recognize six emotions. First, they worked with Ekman set of basic emotions (anger, disgust, fear, joy, sadness, surprise). Then, based on the analysis of texts from various dialogues between the Chatbot and a human, they excluded the emotion "disgust" and replaced it with another emotion, "love." In this new set of emotions, the number of positive and negative emotions is balanced.

According to the literature, CNN and LSTM neural networks achieve the best results in text data processing. In the paper (Ghourabi et al., 2020) following machine learning methods were compared: SVM, K-Nearest Neighbor, Naïve Baes, Decision Tree, Logistic Regression, Random Forest, AdaBoost, Bagging classifier, CNN, LSTM, and a combination of CNN+LSTM. The best results were achieved by CNN+LSTM. Findings from this research are consistent with that paper.

They have provided experiments also with the Lexicon-based approach (LBA), Naïve Bayes (NB), and SVM using BOW representation, for comparison with the neural networks model. The results of classic methods of machine learning are poor. Results showed that the best model is the neural networks model (combined Conv1D+LSTM).

2) Article nr 2: Emotion Classification in Short English Texts using Deep Learning Techniques: [2]

The research article investigated emotion classification in short English texts through a variety of statistical and deep learning methodologies. A meticulously curated dataset of short English texts, annotated with five distinct emotion categories, was employed to support the study. The evaluation of both shallow and deep learning models on the SmallEnglishEmotions dataset underscored the superior accuracy of deep learning architectures, particularly those harnessing semantic embedding and transfer learning techniques, in adeptly classifying emotions. Notably, the utilization of the pretrained distilBERT model in the deep learning framework emerged as the most effective strategy for achieving remarkable performance in English emotion recognition within short texts. Moreover, a comparative analysis of model performance on the SmallEnglishEmotions and a standard English dataset revealed a heightened accuracy and Macro-F1 metrics in classifying shorter texts compared to longer ones.

Various iterations of BERT and similar models have been employed for text embedding purposes. Moreover, transfer learning techniques have been applied to enhance the comprehension of emotional word semantics within deep learning frameworks. Researchers also utilize approaches like combining models from diverse linguistic backgrounds and employing data augmentation techniques to enhance the accuracy of emotion recognition systems. In the domain of English text emotion recognition, early studies primarily relied on shallow learning approaches but gradually transitioned to more sophisticated methods like GRU and BERT-based embedding models such as XLM-R.

3) Article nr 3: Emotion Recognition on Social Media Using Natural Language Processing (NLP) Techniques: [3]

This paper provides an overview of the architecture for three of the most popular Transformer-based models, BERT Base, DistilBERT, and RoBERTa. These models are also fine-tuned using the "Emotions" dataset — a data corpus composed of English tweets annotated in six different emotions, and the performance of the models is evaluated. The results of this experiment showed that while all of the models demonstrated excellent emotion recognition capabilities by obtaining over 92% F1-score, DistilBERT could be trained in nearly half of the time compared to the other models. Thus, the use of DistilBERT for emotion recognition tasks is encouraged. DistilBERT outperformed BERT and RoBERTa in terms of accuracy, F1-score and samples tested per second. Despite the margin of improvement of DistilBERT over the other two models being very small in every metric, it is still considered relevant as the three models have very similar architectures. The RoBERTa model also offered slightly better accuracy results over BERT Base, but their overall F1-score was the same. It is also worth noting that the F1-score is the most relevant performance metric for this experiment and provides the best overview of model performance. This is because F1-score combines both precision and recall and thus takes into account how the data is distributed through the dataset. This is a key observation since the dataset used for the model's training presents a very uneven distribution of labels.

4) Article nr 4: An Emotion Classification Scheme for English Text Using Natural Language Processing: [4]

For the experiment, they have compared the performance of two different state-of-the-art models such as BERT and ELECTRA which are already pre-trained with the Wiki dataset, through fine-tuning and evaluating with rich emotion literature datasets in order to figure out a better model for emotion classification. Bert has a representation extracted through the MLM technique, and Electra, another model, has a representation extracted through the RTD technique. Using data collected from drama scripts, novels, and poems and labeled with the help of a liberal arts department, they contributed to the classification power of the model's reading comprehension ability and emotional understanding. Components are: A literature dataset consisting of 5 classes for fine-tuning, a tokenizer that is an embedding layer that transforms data into a trainable datatype, two SOTA models, a classifier that classifies the 5 output logit through softmax, and F1 confusion matrix that shows the performance and evaluation of the model.

As a result of observation, Bert's training convergence speed was faster than Electra in the same sampling, but the Electra model had a slower learning speed compared to the Bert model, while the accuracy was higher than that of the relatively stable Bert model. In addition, in the Electra model, overfitting was observed at the 10th epoch point. This shows that the RTDbased Electra model has higher model complexity than the MLM-based BERT model. Electra is a more sophisticated model than Bert, and as proof, the F1 score matrix shows that Electra was higher than Bert's f1 score. Also, after merging the data of two classes with high similarity, heartbreaking, and sad classes, the experiment was conducted by re-sampling with the same number of other classes. As a result of observation, based on the training accuracy of the Bert model, the convergence speed of Bert was faster than that of Electra. However, as a result of checking the validation accuracy, Bert showed stable results, while Electra's continuous performance increased at the utmost 60 percent accuracy was confirmed. On top of this, by merging the two classes with high similarity, the understanding of the input sample of Electra was further increased, and it can be confirmed that the predicted values for all classes of Electra in the F1 score have increased than Bert.

5) Article nr 5: An Optimized Deep Learning Model for Emotion Classification in Tweets: [5]

In this paper, hybrid combination of different model's LSTM-CNN have been proposed where LSTM is Long Short Term Memory and CNN represents Convolutional Neural Network. The main drawback of LSTM is that it's a time-consuming process whereas CNN do not express content information in an accurate way, thus our proposed hybrid technique improves the precision rate and helps in achieving better results. For this study, a proper benchmark of the different models is being set and the dataset incorporated is SemEval Datasets which is an ongoing semantic evaluation of computer semantics and has widely been used for bench-marking sentiment analyses. In our research paper, CNN as well as LSTM is being used as both of them individually comprises of certain shortcomings which can be mitigated by using both of them in tandem and this is done by using a hybrid model as mentioned in this paper. CNN has been used to create a pool layer that is further passed to the LSTM model down the pipeline. Another hurdle that this paper tends to solve is the extensive use of emoji's that has drawn growing attention from research and these emoji's contain important information as previous work has shown that, it is useful to pre-train a deep neural network on an emoji prediction task with pre-trained emoji to predicting emotion and sarcasm with greater accuracy. In this work an "opinion lexicon" has been used for a lexicon-based approach that uses punctuations and emoticons as key flags for determination. The main goal was to compare various deep learning and machine learning models. Furthermore, hybrid combination of different model's LSTM-CNN have been proposed to achieve more accuracy and efficiency by reducing latency.

II. Methodology

A. Dataset

For emotion classification we used a dataset, which have a collection of English Twitter messages annotated with six fundamental emotions: anger, fear, joy, love, sadness, and surprise, available on kaggle. The plot below (Fig. 1) displays the distribution of these categories in the dataset, providing a visual representation of the frequency of each emotion:

B. Data Exploration and Analysis

In this subsection, we explore and analyze the dataset by visualizing the distribution of sentiments, the distribution of tweet lengths, and the most frequent words associated with different emotions. These visualizations help in understanding the characteristics of the data and the common expressions related to various emotions.

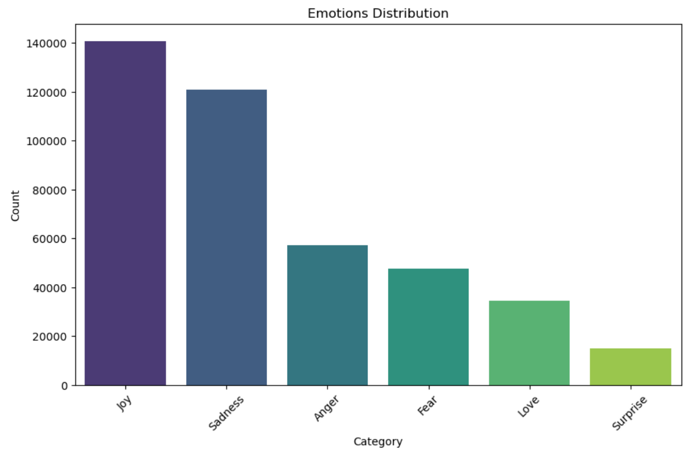

Fig. 1: Categories distribution.

Figure 2 shows the distribution of different sentiments across three categories: Neutral, Positive, and Negative. The counts for each sentiment are represented by different colors in the bar plot.

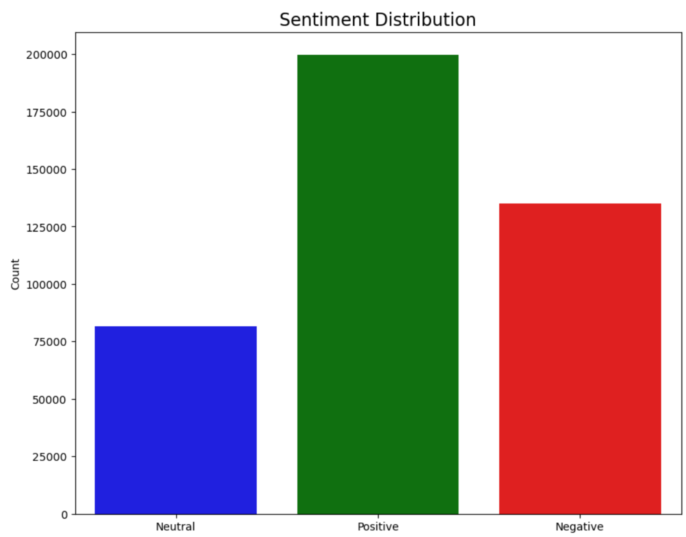

Fig. 2: Sentiments Distribution

Figure 3 shows the distribution of tweet lengths in the dataset. This histogram indicates the frequency of tweets of varying lengths, which are used to define emotions.

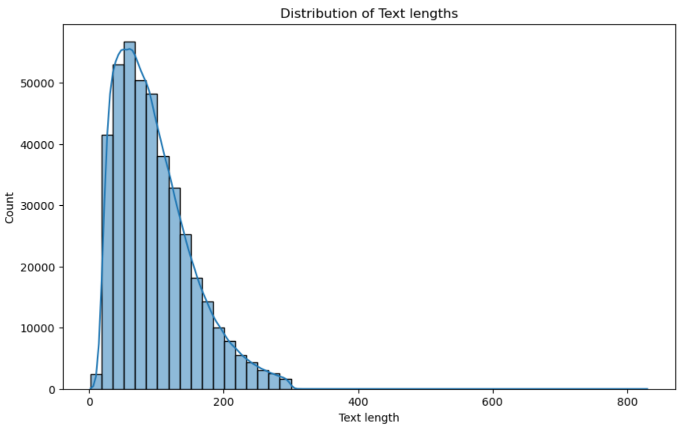

Fig. 3: Distribution of tweets length

Figure 4 shows the word clouds for various emotions in the dataset, including fear, sadness, love, anger, joy, and surprise. Each word cloud highlights the most common words associated with each emotion, with the size of the words indicating their frequency.

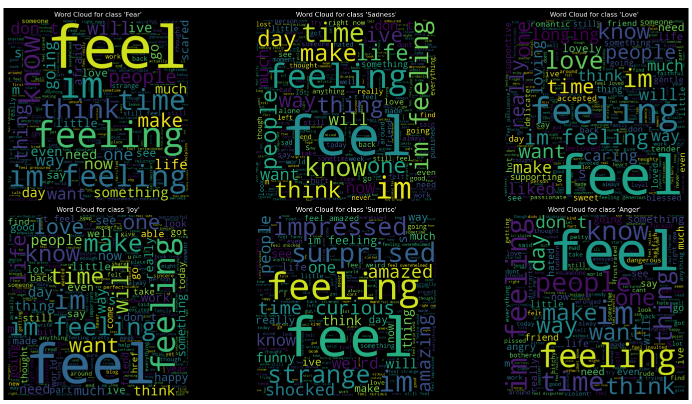

Fig. 4: Word Clouds

C. Initial Data Preprocessing

The initial step in our methodology involves preparing the tweet data for further analysis and model training. The preprocessing steps are crucial for reducing noise and ensuring the model focuses on relevant features. The following steps are implemented in Python:

1) **Text Normalization:** All text data is converted to lowercase to ensure uniformity, minimizing the variability introduced by capitalization.
2) **Stopword Removal:** Common English stopwords are removed using the NLTK library. This helps in reducing the size of the data and focuses on more meaningful words.
3) **Punctuation Removal:** All punctuation marks are stripped from the text, as these do not contribute to the sentiment of the text.
4) **Digit Removal:** Numerical digits are removed since they generally do not carry emotional information.
5) **Whitespace Normalization:** Extra spaces are removed to clean up the text.

The cleaned text data is then stored in a new column within our dataset, ensuring we maintain a distinction between raw and processed data. This process is encapsulated in the `data_cleaning` function applied to the ‘text‘ column of our DataFrame.

Post-preprocessing, the text data is tokenized. Tokenization involves converting text into a sequence of integers, where each integer represents a unique word in a dictionary. This step is necessary for preparing the input to neural network models which require numerical input.

1) A tokenizer is fitted to the cleaned text, mapping each unique word to an integer.
2) Text sequences are padded to ensure uniform input size for modeling. The maximum sequence length is determined by the longest tweet post-cleaning.

We employ the pre-trained GloVe embeddings to convert text tokens into embedding vectors. GloVe embeddings are chosen due to their ability to capture syntactic and semantic word relationships developed from a large corpus of text.

1) An embedding matrix is created where each row corresponds to a token’s embedding vector, indexed by the token’s integer representation from the tokenizer.
2) Only words present in our tokenizer and the GloVe corpus are considered, ensuring our model learns from relevant pre-trained vectors.

Finally the dataset is split into training and testing sets, with 80% of the data used for training and the remaining 20% used for evaluation. This split ensures that the model is tested on unseen data, evaluating its ability to generalize.

With the preprocessed and vectorized data, the model is trained to classify emotions. This involves configuring the neural network architecture, setting training parameters, and initiating the training process using the training data subset what we desctribe in the following sections.

D. Baseline Model

The baseline model for our study is Logistic Regression, which is extensively employed in statistical modeling of binary and multi-class classification problems. Due to its robustness and efficiency in handling linear relationships between the dependent and independent variables, Logistic Regression serves as an ideal choice for initial benchmarks in text classification tasks.

Logistic Regression estimates the probabilities using a logistic function, which is the cumulative logistic distribution. In the binary classification framework, this model predicts the probability that a given input belongs to a category, labeled ’1’ as opposed to ’0’. For multi-class classification, the approach is generalized to multiple logistic regression, also known as multinomial logistic regression, where the logistic function is replaced with a softmax function across the multiple classes.

The training process involves optimizing the parameters of the logistic model to minimize the cost function, typically the logarithmic loss, across the training data. This process adjusts the weights applied to features in order to reduce the prediction error between the actual class labels and the predictions made by the model.

The performance of Logistic Regression is evaluated based on metrics such as accuracy, precision, recall, and F1-score. It will help us verify whether Deep learning solutions perform better than more traditional approach.

TABLE I: Summary of Logistic Regression Performance Metrics

Class	Precision	Recall	F1-Score
sadness	0.94	0.93	0.94
joy	0.92	0.92	0.92
love	0.77	0.76	0.76
anger	0.90	0.89	0.90
fear	0.84	0.84	0.84
surprise	0.70	0.72	0.71
Overall Metrics			
Accuracy		0.89	
Macro Avg	0.84	0.84	0.84
Weighted Avg	0.89	0.89	0.89

Table I shows a comparison of the metrics according to predicted grade, as well as their generalised results. As can be seen, the general accuracy metric oscillates around 89%, which is a good result for a baseline model. Slightly lower values (84%) are seen for statistics such as Precision, Recall or F1-Score. The model also suffers from underfitting when it comes to class love. This can be seen well on the confusion matrix (figure 5), where this class is often indistinguishable from the joy class from the model's perspective. This is evidence that there is room to improve the model with Deep learning methods, although it should be noted that the baseline model stood out for its relatively high acccuracy and it is certainly a much faster method than machine learning models based on Neural Networks.

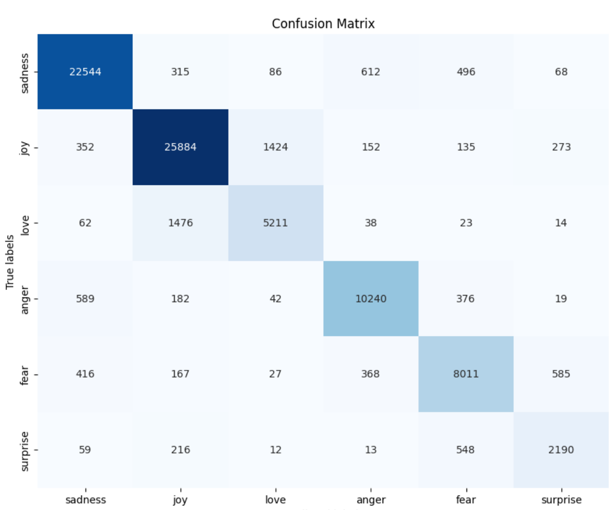

Fig. 5: Confusion Matrix for Baseline Model

E. Model Architecture

Our main model is neural network designed for text classification that leverages both pre-trained word embeddings and recurrent neural network components. Here is a breakdown of each component of the architecture:

1) **Embedding Layer:** The first layer is an embedding layer that uses GloVe pre-trained word vectors. The dimensions of the embedding are set to 300 to match the GloVe embeddings. This layer transforms the integer-encoded tweets into word vector representations. The embedding layer's weights are initialized with the GloVe matrix and are set to be non-trainable to preserve the semantic properties learned from the large corpus on which GloVe was trained.
2) **Input Layer:** The input layer takes sequences of integers (tokenized text) of fixed length (79 tokens after padding).
3) **Bidirectional LSTM Layer:** A bidirectional LSTM with 64 units is used to process the word embeddings. This LSTM layer can capture context from both the past and future states, allowing it to better understand the sequence context. Dropout and recurrent dropout of 0.2 are applied to prevent overfitting by randomly ignoring certain features during training.
4) **Output Layer:** The output from the bidirectional LSTM is passed to a dense layer with 6 units, corresponding to the six emotion classes. This layer uses a softmax activation function to output a probability distribution over the six classes, where each probability indicates the likelihood of a given class being the true class for the input tweet.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 79)]              0

 embedding (Embedding)       (None, 79, 300)           22545300

 bidirectional (Bidirectiona  (None, 128)              186880
 l)

 dense (Dense)               (None, 6)                 774

=================================================================
Total params: 22,732,954
Trainable params: 187,654
Non-trainable params: 22,545,300
```

Fig. 6: Summary of the Model Architecture

The model is compiled with the Adam optimizer, a common choice for deep learning applications due to its efficient handling of sparse gradients on noisy problems. The loss function used is categorical crossentropy, which is suitable for multi-class classification problems. Accuracy is used as the metric for model performance evaluation.

F. Training

When training the model, in addition to the traditional structure described in the previous chapter, we introduced two an augmentation technique to improve model performance.

We used is synonym augmentation that is a technique where words in a text are replaced with their synonyms. It relies on maintaining the semantic meaning of the text while altering its lexical composition. This method is particularly useful for tasks that require the model to understand and process text at a semantic level rather than just recognizing specific word patterns.

The steps involved in synonym augmentation are:

1) Selecting words in the text that are suitable for replacement, typically avoiding function words that are crucial for grammatical structure.
2) Identifying suitable synonyms for the selected words using a thesaurus or lexical database such as WordNet.
3) Replacing the original words with their synonyms to create a new version of the text.

Finally we conducted two different model, what is

1) LSTM model based on non-augmented data.
2) LSTM model based on augmented (synonym augmentation) data.

In the next section we compare results obtained from these approaches.

III. Experiments and Results

Let us therefore move on to the experimental part, which will focus on the selection of the final model, which we will compare with other methods described in recent literature.

A. Evaluation

We first built a model, on a reprocessed but unbalanced dataset. The results can be found in Table II.

TABLE II: Classification Performance Metrics for LSTM non-augmented model

Class	Precision	Recall	F1-Score
sadness	0.98	0.97	0.97
joy	0.97	0.93	0.95
love	0.79	0.93	0.85
anger	0.95	0.93	0.94
fear	0.91	0.88	0.90
surprise	0.75	0.94	0.83
Overall Metrics			
Accuracy		0.94	
Macro Avg	0.89	0.93	0.91
Weighted Avg	0.94	0.94	0.94
F2-Score		0.92	

As can be seen, most of the metrics went up compared to the baseline model led by an accuracy of 94%. The underfitting concerning the love class is no longer so significant, but it still seems to be an element to be refined (Precision at 79%). The same conclusions can be drawn from figure 7, which shows the confusion matrix of the model.

Let's look further at the process of training the model (figure 8), actually changing the value of the loss function (which we want to minimise) and the accuracy (which we want to maximise). As you can see, around epoch 4-5, both functions seem to stabilise their value, indicating that the number of epochs has been chosen appropriately. There are also no particular signs of under or over-fitting.

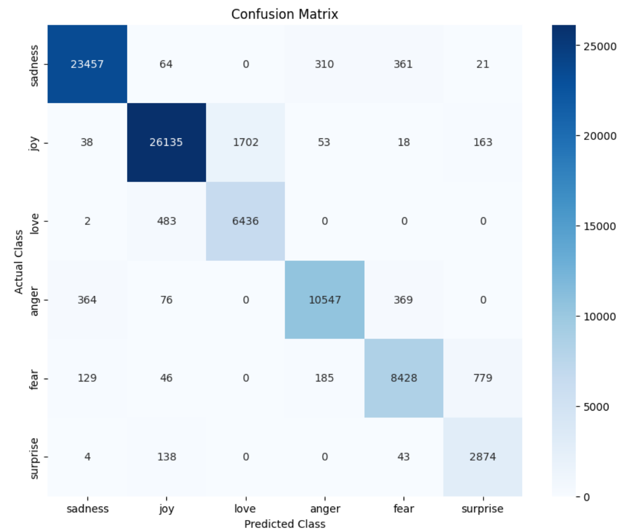

Fig. 7: Confusion Matrix for LSTM model based on non-augmented data

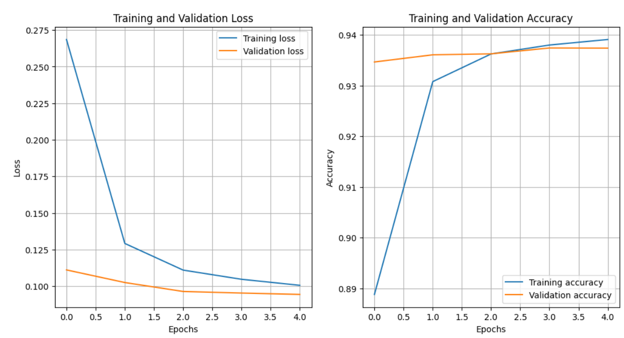

Fig. 8: Loss and Accuracy plots for LSTM model based on non-augmented data

Let us now turn to the description of the result based on augmented data. After data balancing (significant augmentation of the set by synonymised replication of samples), the metrics for all classes stabilised, in particular for the love class, which previous solutions suffered from (table III).

The same conclusion can be reached by analysing the confusion matrix (figure 9).

TABLE III: Classification Performance Metrics for Augmented Data

Class	Precision	Recall	F1-Score
sadness	0.94	0.94	0.94
joy	0.97	0.90	0.93
love	0.90	0.98	0.94
anger	0.91	0.94	0.93
fear	0.96	0.83	0.89
surprise	0.90	0.98	0.94
Overall Metrics			
Accuracy		0.93	
Macro Avg	0.93	0.93	0.93
Weighted Avg	0.93	0.93	0.93
F2-Score		0.93	

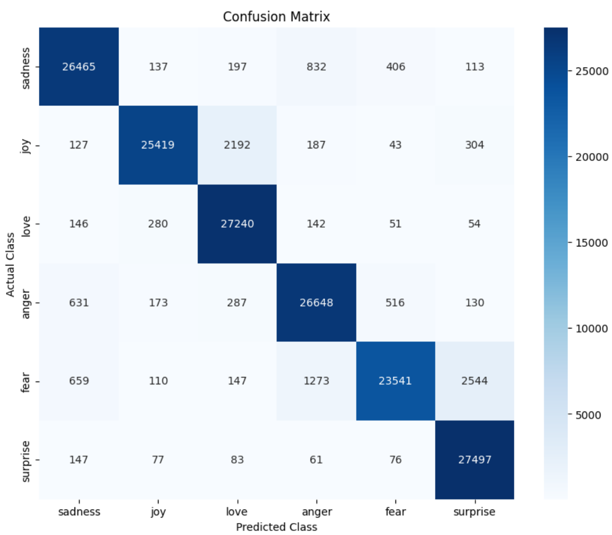

Fig. 9: Confusion Matrix for LSTM model based on augmented data using synonym augmentation technique

On the other hand, as can be seen from the graph 10 showing the values of the loss function and accuracy, no consesus was found between the training set and the validation set. Interestingly, better metrics can be seen on the validation set, which may indicate underfitting of the model. One could conclude that the number of epochs should be increased. However, firstly, this would significantly increase the training time (which was already increased compared to the previous approach) and secondly, the graphs seem to flatten out around epoch five, which is not a trend that allows for optimism in this solution

B. Comparison to the other articles

Table IV presents our LSTM models' performance metrics compared to those from recent studies on emotion recognition in text data.

The non-augmented LSTM model outperforms other approaches with the highest accuracy (0.94) and a strong F1-Score (0.91). The augmented LSTM shows slightly reduced accuracy but enhanced precision and recall (both 0.93), suggesting that data augmentation benefits model generalization.

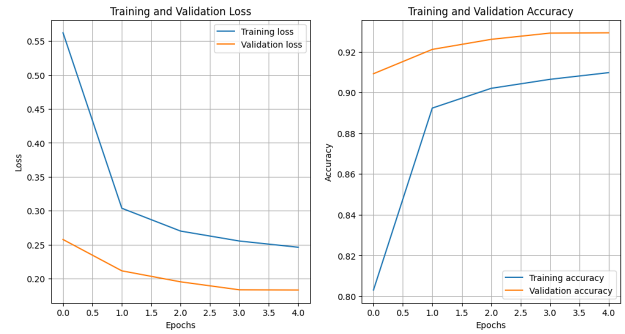

Fig. 10: Loss and Accuracy plots for LSTM model based on augmented data using synonym augmentation technique

TABLE IV: Comparison of Classification Metrics Across Various Studies

Article	No Classes	Accuracy	Precision	Recall	F1-Score
[1] – LSTM + CNN	2	0.86	0.90	-	0.88
[2] – DistilBERT	6	0.93	-	-	0.93
[3] – DistilBERT	5	0.77	-	-	0.73
[5] – LSTM + CNN	6	0.91	0.87	0.87	0.87
LSTM non-augmented	6	0.94	0.89	0.93	0.91
LSTM augmented	6	0.93	0.93	0.93	0.93

These results demonstrate the effectiveness of LSTM architectures, which remain competitive with advanced models like DistilBERT, particularly in handling multiple classes with high precision and recall balance.

IV. Conclusion

This study demonstrated the efficacy of LSTM-based models, both non-augmented and augmented, in the emotion classification task on text data. Our models not only performed competitively with state-of-the-art methods like DistilBERT but also showed superior performance in certain metrics. Specifically, the augmented LSTM model achieved remarkable balance in precision and recall, suggesting robustness in generalization across various emotional contexts. These results underline the potential of LSTM models to handle complex, nuanced tasks such as emotion recognition in textual data, offering a viable alternative to more computationally expensive transformer models.

V. Future Work

Future research will extend in several strategic directions to enhance the performance and applicability of our models. Firstly, we plan to explore the integration of context-aware embeddings such as BERT and RoBERTa, which may deepen the models' contextual understanding and potentially improve classification accuracy. Additionally, diversifying our training datasets to include text from various sources like forums and blogs will be crucial to testing the robustness of our models across different linguistic styles and contexts.

A significant future endeavor will involve adapting our models for real-time emotion recognition suitable for practical applications such as live customer service interactions or social media monitoring. This real-time capability could transform customer experience management and social media analytics by providing immediate insights into user sentiments.

Interdisciplinary approaches that combine insights from psychology and linguistics will also be considered to refine the models' ability to interpret emotional nuances in text. This could lead to more sophisticated classifications and enhanced training strategies. Finally, it is imperative to address the ethical considerations related to privacy and the psychological impacts of emotion detection. Ensuring that applications of this technology respect user consent and data protection laws will be paramount to their ethical deployment in real-world scenarios.

These initiatives aim to not only advance the technical capabilities of emotion recognition systems but also to ensure their ethical and practical viability for deployment in diverse applications.

References

[1] K. Machová, M. Szabóova, J. Paralič, and J. Mičko, "Detection of emotion by text analysis using machine learning," *Frontiers in Psychology*, vol. 14, p. 1190326, 2023. DOI: 10.3389/fpsyg.2023.1190326. Available: https://www.frontiersin.org/articles/10.3389/fpsyg.2023.1190326/full.

[2] S. Bhat, "Emotion Classification in Short English Texts using Deep Learning Techniques," *arXiv*, Feb. 2024. Available: https://arxiv.org/html/2402.16034v2.

[3] L. Gomez, T. Watt, K. Babaagba, C. Chrysoulas, A. Homay, R. Rangarajan, X. Liu, "Emotion Recognition on Social Media Using Natural Language Processing (NLP) Techniques", ICISS 2023, August 11–13, 2023, Edinburgh, United Kingdom DOI: 10.1145/3625156.3625173. Available: https://dl.acm.org/doi/fullHtml/10.1145/3625156.3625173.

[4] M. Gu, J. Kwon, J. Jeong, S. Kwon "An Emotion Classification Scheme for English Text Using Natural Language Processing"

[5] X. Zhang, Y. Liu, "An Optimized Deep Learning Model for Emotion Classification in Tweets," *CMC: Computers, Materials & Continua*, vol. 70, no. 3, pp. 4515-4533, 2022. DOI: 10.32604/cmc.2022.013695. Available: https://www.techscience.com/cmc/v70n3/44965/html.