Emotion Detection from Plain Text

Hongxin Song, Ceci Chen, Lia Wang, Maggie Xu
DS-GA 1003 Machine Learning

Abstract

The detection of emotions in textual data is critical for applications such as mental health monitoring, digital content moderation, and consumer behavior analysis. Traditional methods like Support Vector Machines and Naïve Bayes, and other classical machine learning algorithms, while foundational, often fall short in capturing the full spectrum of human emotions due to their limitations in contextual understanding. This study introduces the use of **BERT** (Bidirectional Encoder Representations from Transformers), a leading-edge model that utilizes masked language modeling and next-sentence prediction, to enhance emotion detection capabilities in text (2)(Devlin et al., 2018). Our methodology follows a structured process, starting from data collection and prepossessing to modeling with both traditional algorithms and BERT and culminating in a comprehensive evaluation of model performance. Preliminary results indicate that BERT significantly outperforms baseline models, offering a promising avenue for more nuanced and accurate emotion analysis in textual data. This research not only advances the technical understanding of emotion detection but also proposes practical implications for improving human-machine interactions.

1 Introduction

Emotional detection in text is a crucial field that helps businesses, governments, and various organizations gauge public sentiment and refine their strategies. Research on emotional detection in the text has evolved significantly, transitioning from early rule-based systems using sentiment lexicons to sophisticated machine-learning methods. The introduction of transformer-based models like BERT further advanced the field, enhancing the understanding of context and textual nuances. Developed by researchers at Google, BERT processes words in relation to all the other words in a sentence, rather than one-by-one in order. This allows the model to interpret the full context of a word by looking at the words that come before and after it—making it particularly effective for understanding the intent behind search queries and the sentiment of sentences.

This study aims to harness the contextual processing power of BERT to enhance emotion detection in textual data. By integrating BERT with traditional approaches, we anticipate our model to not only achieve higher accuracy but also to be adept at interpreting complex emotional nuances that previous models may overlook. This integration promises to refine the precision of emotion detection and extend its applicability to more complex scenarios.

1.1 Data

The raw dataset used for this project is a text dataset extracted from Twitter posts finding from Kaggle(3), which is a collection of tweets annotated with the emotions behind the content. The raw datasets contain 40,000 records in total and have the following 3 columns:

- "tweet_id": Twitter user ID in integer type
- "sentiment": 13 pre-defined emotion classes assign to content in string type

Preprint. Work in progress.

- "content": raw tweets content in string type

The figure 1.1 below shows the number of records for different sentiment classes:

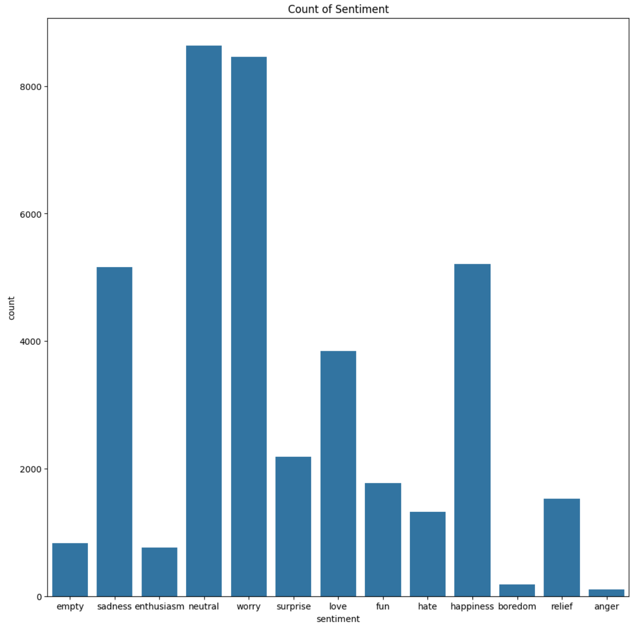


Figure 1.1

2 Related Work

In previous studies, researchers have explored various approaches to detect emotions from text. In one of the studies, researchers utilized various algorithms such as Logistic Regression, Linear Support Vector Machine, and Random Forest to categorize emotions from text, and their comparative analysis focused on two key features: Term Frequency-Inverse Document Frequency (TF-IDF) and Count Vectors (6)(Singh et al., 2023). These features have limitations in textual analysis due to the single-directional nature of the traditional models: the analysis is not based on full context. The field of Textual Emotion Detection (TED) also is crucial in health and medicine researches. To better understand patient sentiments and reactions for diagnosing mental health conditions like depression and monitoring the emotional well-being of patients with chronic diseases, the relevant studies mark the shift towards deep learning techniques. This shift proves that more advanced models have improved accuracy over traditional methods.(1) (Saffar et al., 2023). Our study aims to employ a state-of-the-art model to achieve improvements in the classification performance of emotion detection from text.

3 Methodology

3.1 Data Pre-possessing

The data cleaning started with checking if there were any rows with null values or any duplicated rows, of which there are none in this dataset. Then, drop the "tweet_id" column to make sure only two columns ("sentiment" & "content") useful in modeling are kept. To further unify the context format, the following actions are taken:

1. Replacing abbreviated forms of words or syllables with their expanded, full-word forms in a given string of text.
2. Removing Twitter username mentions, which are identified by "@" followed by a word and optional white space.

3. Converting all characters in the text to lowercase to standardize the input.
4. Removing links that start with "`http`" or "`https`", and links starting with "www." and ending with ".com".
5. Cleaning out special characters or symbols that do not reflect emotions(! ? excluded).
6. Splitting text into a list of words based on white space.

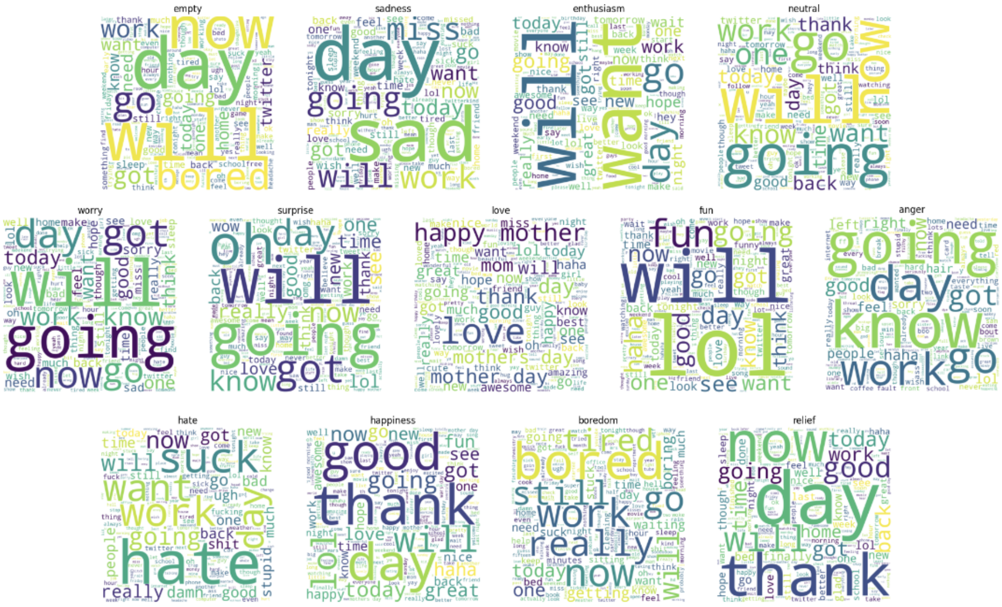


Figure 3.1 : The Wordcloud figure shows the frequency of words in each sentiment category after finishing the initial data-cleaning step. The larger the size of a word is, the more frequently it appears in the specific sentiment class.

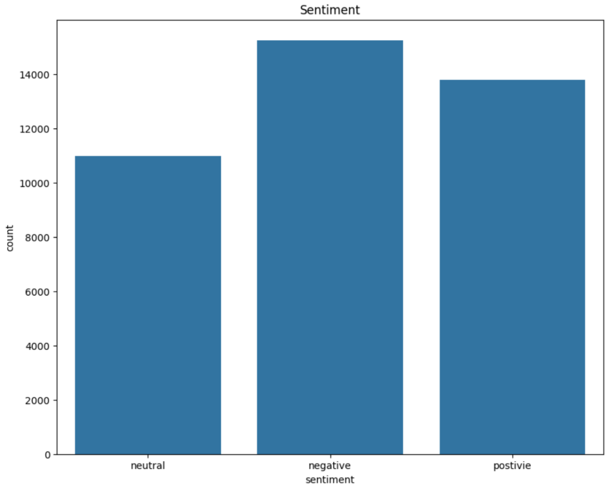


Figure 3.2 Data Classes after Pre-processing

To better balance the class distribution and help handle possible noise in the dataset, we have categorized the 13 sentiments into 3 main categories before entering the modeling part: positive, negative, and neutral. The **positive** category contains sentiments of **"happiness"**, **"enthusiasm"**, **"surprise"**, and **"love"**. The **negative** category contains sentiments of **"boredom"**, **"hate"**, **"sadness"**, **"anger"**, and **"worry"**. The **neutral** category contains sentiments of **"relief"** and **"empty"**. The distribution of the processed classes is displayed in Figure 3.2. We can tell the differences between classes are much smaller compared to the raw dataset.

3.2 Modeling

In this study, we began our empirical investigation by establishing a baseline with traditional machine learning models: Decision Trees, Random Forests, and Logistic Regression. These models were selected due to their widespread use and straightforward implementation. However, they often exhibit significant limitations when addressing the complexities inherent in natural language processing (NLP) tasks, particularly due to their inability to capture contextual relationships within text data. We chose the Decision Tree as our Baseline Model, for it resulted in the lowest level of accuracy after training.

3.2.1 BERT

To address these shortcomings, we incorporated the Bidirectional Encoder Representations from Transformers (BERT) model, renowned for its effectiveness in a wide range of NLP applications. BERT is a pre-trained transformer model designed to handle contextual relationships in text by pre-training on a large corpus using tasks like masked language modeling (MLM) and next-sentence prediction.(2)(Devlin et al., 2018) This pre-training is followed by fine-tuning particular downstream tasks, such as sentiment analysis or named entity recognition.

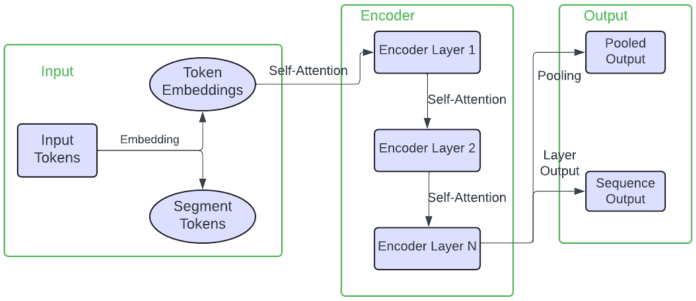


Figure 3.3 BERT Architecture Diagram

In the beginning, BERT takes in text that has been tokenized into "Input Tokens." This tokenization splits the text into pieces, such as words or sub-words to process different elements of language. Once we have our input tokens, they move into the embedding stage, where each token is converted into a vector known as "Token Embedding." To understand sentence relationships, BERT uses "Segment Tokens." This transformation enables the model's encoder to work.

The encoder, which is the core part of BERT, consists of multiple layers, which represents the depth of the model. Each layer includes a self-attention mechanism, which can evaluate and understand the context of each token in relation to others in the sentence so that BERT can better understand the meanings based on the context provided by other words.

Finally, BERT produces two main types of outputs. The "Pooled Output" is a consolidated representation of the entire input sequence, typically used for tasks like sentiment analysis or classification. The "Sequence Output" provides a detailed representation for each token, which is useful for tasks like named entity recognition or question answering.

3.2.2 Fine Tuning

Fine-tuning the BERT model involved systematically optimizing several hyper-parameters to tailor the model to our specific dataset and task requirements. We conducted both manual parameter tuning and Grid Search to find the optimal setup. Specifically, we first adjusted the learning rate, dropout rates, batch size, and the number of training epochs manually. After we determined the number of epochs, we further tuned the parameters by performing a Grid Search for the learning rate, batch size, and weight decay. Below is a more detailed description of our parameter-tuning process and reasoning.

1. **Learning Rate**: We experimented with various learning rates (1e-5, 2e-5, 5e-5 for manual tuning and then 1e-5, 2e-5 for Grid Search) to identify the rate that allows for the quickest convergence without leading to instability in the training process.
2. **Dropout Layers**: To combat the risk of over-fitting—common in deep learning models with a large number of parameters—we integrated dropout layers. These layers randomly omit a subset of features at each iteration during the training phase, which helps to improve the generalization of the model.
3. **Batch Size**: Adjusting the batch size provided insights into how the model's performance could be balanced with computational efficiency. Smaller batch sizes often lead to more noise in the gradient estimates, but larger batch sizes could hinder the model's ability to generalize. We tried 16, 64, 128 for our manual tuning and 16, 64 for the Grid Search process.
4. **Weight Decay**: To further avoid over-fitting and stabilize the model's performance, we added weight decay as a new parameter for tuning. We experimented with 0.01 and 0.001 for the Grid Search process, and we also monitored the interaction between weight decay and learning rate to yield better results. A lower learning rate might require a different weight decay value to balance the regularization effect.
5. **Number of Epochs**: The optimal number of epochs was determined to ensure that the model was adequately trained over the dataset, without excessive training leading to over-fitting. This was done by monitoring the model's performance on a validation set and stopping training when performance plateaued.

This systematic approach to hyper-parameter tuning not only enhanced the performance of our BERT model on the task at hand but also ensured that the model was robust enough to generalize well to new, unseen data.

4 Results

As shown in Table 1, we tested three initial models: Decision Tree, Logistic Regression, and Random Forest. The Decision Tree model yields a test accuracy of about 42.51%, Logistic Regression performed the best out of three models with an accuracy of 57%, and Random Forest was close with an accuracy of 56.4%.

Table 1 Comparison of Model Accuracy

Model	Test Accuracy
Baseline (Decision Tree)	42.51%
Logistic Regression	57%
Random Forest	56.4%
BERT w/ fine-tuning	**63.18%**

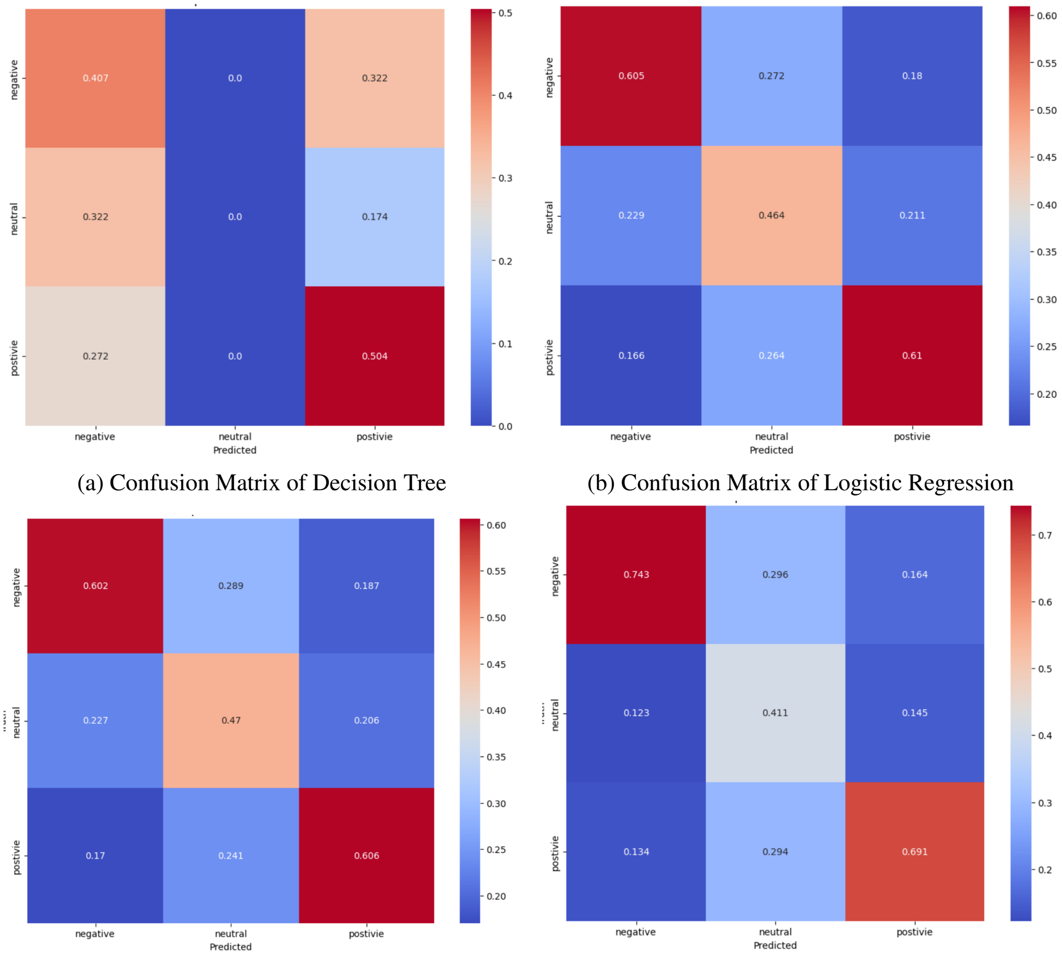
We trained the BERT model with Grid-Search, testing different hyper-parameters including learning rate, batch size, weight decay, and drop-out rate. The final BERT model after Grid-Search has a learning rate of $1 * e^{-}5$, batch size of 64, drop-out rate of 0.1, and weight decay rate of 0.01. With thorough experiments, we discovered that for large language models like BERT, 3-5 epochs of training are enough for them to learn from the dataset. The model would start to over-fit after a number of epochs. By examining the classification report in Table 2, we discovered that BERT performs best for predicting negative class with an F-1 score of 0.7. The model performs the worst for the neutral class. The recall of the neutral class is only 0.41, which suggests that the majority of the neutral classes are misclassified into other classes.

From the confusion matrix in figure 4.1, all models have the worst performance on the neutral class. The baseline Decision Tree model is not able to predict a neutral class at all. This is probably due to the fact that the neutral class is too ambiguous. The Decision Tree and Random Forest models

Table 2 Classification Report for BERT

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Negative	0.67	0.74	0.70	3003
Neutral	0.55	0.41	0.47	2253
Positive	0.64	0.69	0.66	2744
Accuracy			0.63	8000
Weighted Avg	0.62	0.63	0.62	8000

show a logical performance pattern with good performance on more distinctive classes like negative and positive and worse performance on more ambiguous classes like neutral. Logistic Regression shows a slightly more balanced performance on different classes but it also perform the worst on the neutral classes. BERT has the best performance out of all the models, which can be seen from the figure 4.1d. Although being able to correctly classify most of the negative and positive class, BERT shows the inability of distinguish neutral from the positive class but less so from the negative.



(a) Confusion Matrix of Decision Tree

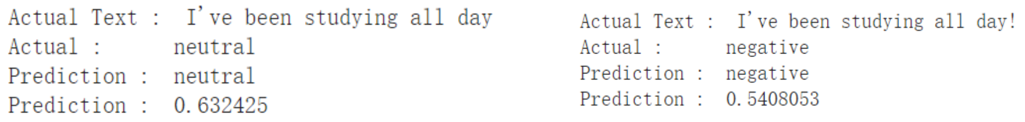
(b) Confusion Matrix of Logistic Regression

(c) Confusion Matrix of Random Forest

(d) Confusion Matrix of BERT after Fine-tuning

Figure 4.1 Comparison of Confusion Matrices Across Different Models

In order to test the model's performance, we experimented with different sentences. The model shows overall good abilities to distinguish negative and positive content. It is interesting how adding a simple special character can change the tone of the sentence. BERT is able to detect these subtle differences displayed in the figure 3.2



```
Actual Text :  I've been studying all day
Actual :       neutral
Prediction :   neutral
Prediction :   0.632425
```

```
Actual Text :  I've been studying all day!
Actual :       negative
Prediction :   negative
Prediction :   0.5408053
```

Figure 4.2 Test of BERT on TEXT

5 Conclusion

In conclusion, BERT outperforms all simple models like decision trees and logistic regression. This shows the ability of BERT to handle complex text data. For large and pre-trained model like BERT, a smaller learning rate typically achieve better results. Smaller batch sizes and smaller learning rate allows the model to learn slowly and subtlely from the training data. Since BERT is already a very refined model of text classification, we would want to let it slowly adapt to new task without changing its pre-trained weights too much. Adding some dropout and tuning batch size would help improve the model's generalization ability. Due to the complexity of BERT, it is easy to over-fit the training data. This makes it crucial to add an early stopping based on performance on the validation set.

All models perform worst for neutral classes which is probably related to the class imbalance as displayed in figure 3.2. This could also caused by the nature of the neutral class, which is ambiguous compared to the definition of positive and negative.

5.1 Limitation

The performance of the model could be limited by the nature of these emotions. For instance, neutral emotion is more general and usually with no obvious indicator and thus hard to classify. The model would naturally perform better on classes with strong indicators. Also, data noise is always a crucial part that can negatively impact a model's ability to learn effectively. Since our data is collected from Twitter, a social media platform, it contains a considerable amount of noise, including typographical errors and slang commonly found in internet-based communications. We need more rigorous data cleaning to combat the problem and refine the data input. For example, it might be helpful to implement automated spell checkers to correct common misspellings, define a comprehensive dictionary to translate slang and abbreviations into their standard forms, and further remove or correct incomplete sentences or excessive punctuation. By addressing challenges in social media language through a more advanced pre-processing method, we can significantly enhance the model's performance.

6 Future Work

6.1 Additional Area for Future Research

Hyperparameter tuning is crucial in optimizing model performance. Currently, we only use standard methods like Learning Rate, Dropout Layers, Batch Size, Weight Decay, and Number of Epochs for hyperparameter tuning in this research study. We would like to explore advanced techniques such as Bayesian optimization, genetic algorithms, or gradient-based optimization, which could potentially enhance our model's performance by finding better parameter settings more efficiently.

While transformer models like BERT have set benchmarks in natural language processing, newer or modified architectures could offer improvements. For instance, models like RoBERTa(5)(Liu et al., 2019), which modifies the pre-training procedure of BERT for more robust performance and trained specifically on tweet data, or ALBERT(4)(Lan et al., 2019), which reduces model size and increases speed, could be tested. Additionally, experimenting with domain-specific transformers that are pre-trained on text similar to the one used in your dataset might yield better sentiment analysis results.

Another area for improvement is to expanded the training dataset to include a wider variety of text sources and sentiments. This can ensure the model performs well across diverse demographic groups and does not propagate biases. This includes texts from different regions, dialects, demographic groups, and genres. Ensuring a balanced representation of various sentiments across these groups can help in improving the fairness and generalization of the model. Moreover, techniques such as adversarial training, where the model is trained to perform well even on examples designed to mislead it, can further enhance robustness.

6.2 Future Practical Application

Advancing BERT models for emotional detection in text offers exciting opportunities for innovation across multiple sectors. In customer service, enhancing the emotional intelligence of chat-bots could significantly improve user interactions, making them more personalized and empathetic, thereby boosting customer satisfaction. Additionally, in the realm of mental health, these advancements could provide critical support by enabling the early detection of emotional distress from textual analysis on social media or personal communications. This would allow for timely interventions, potentially mitigating mental health crises. Furthermore, in the educational sector, emotionally aware systems could revolutionize learning environments by providing real-time feedback on student engagement and emotional states, thus allowing educators to tailor their interventions more effectively. Each of these applications not only enhances the functionality of existing systems but also opens avenues for new technologies that can make significant impacts on society and business efficiencies.

References

[1] A. H. Saffar, T. K. Mann, and B. Ofoghi, "Textual emotion detection in health: Advances and applications," *Journal of Biomedical Informatics*, vol. 137, 2023, Art. no. 104258, https://doi.org/10.1016/j.jbi.2022.104258.

[2] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018, October 11). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv.org. https://arxiv.org/abs/1810.04805

[3] Gupta, P. (2021). *Emotion Detection from Text* [Data set]. data.world. https://www.kaggle.com/datasets/pashupatigupta/emotion-detection-from-text/data

[4] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019, September 26). ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv.org*. Retrieved from https://arxiv.org/abs/1909.11942

[5] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019, July 26). ROBERTA: A robustly optimized BERT pretraining approach. *arXiv.org*. Retrieved from https://arxiv.org/abs/1907.11692

[6] V. Singh, M. Sharma, A. Shirode, and S. Mirchandani, "Text Emotion Detection using Machine Learning Algorithms," *2023 8th International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, 2023, pp. 1264–1268, doi: 10.1109/ICCES57224.2023.10192783.