Emotion Recognition in Online Communication

Tommy Zhu

Suvass Ravala

Augustin Kim

August 4, 2025

Contents

1 - Abstract

Understanding emotion in online text is essential for improving communication and reducing misinterpretation in digital interactions. This project evaluates several machine learning models for detecting emotions in tweets, including Naive Bayes, Deep Averaging Networks, Convolutional Neural Networks, and BERT. Each model is trained and tested on a labeled dataset of tweets and assessed using metrics that account for class imbalance. In addition to comparing overall performance, we analyze the types of errors made by each model to identify common challenges, such as confusion between similar emotions and the effects of ambiguous language. Our findings highlight the value of advanced model architectures for emotion recognition and suggest future directions for dialing in sentiment analysis.

2 - Introduction

From back-and-forth conversations on social media with friends to professional exchanges over email, communicating online has become a standard part of our daily lives. However, without human cues like tone or body language, it can be difficult to interpret how someone feels from just words on a screen. Nuances are often lost in electronic conversations, and casual online language tends to be more indirect or ambiguous, making true emotions harder to detect. Accurate emotion detection in language helps prevent miscommunication and the social friction that can result from misinterpreted tone. To establish a meaningful baseline for comparison, we use a Naive Bayes classifier as our initial model for emotion detection. We then evaluate the performance gains from more advanced models: BERT, Deep Averaging Network (DAN), and Convolutional Neural Network (CNN). Each model is trained on tokenized and preprocessed tweet data, allowing us to compare how these architectures handle the informal and unpredictable language often found in online communication. Our main objective is to determine which model achieves the highest accuracy in predicting emotions from tweets, while also identifying the specific areas where each model falls short. The baseline Naive Bayes model achieved a relatively low accuracy of 82%. More advanced models saw significant improvement in performance, with DAN achieving an accuracy of 86% and macro-averaged F1-score of 86%. CNN showed similar improvement, achieving an accuracy of 91% and a macro-averaged F1-score of 88%. The best model, BERT, reached an accuracy of 93% and a macro-averaged F1-score of 93%.

3 - Project Overview

3.1 - Datasets used

The dataset used for this project is the Emotions Dataset for NLP (Kaggle Dataset). It consists of sentences each labeled with one of six emotions: anger, love, fear, joy, sadness, or surprise. The data is provided in CSV format, with each row containing a pre-processed sentence and its corresponding emotion label split by a semicolon.

3.2 - Background and problem approach

Previous research has taken important steps toward recognizing emotions in text, exploring both graph based and probabilistic approaches to better capture how emotions are expressed in text. The CARER framework (Saravia et al., 2018) uses a graph based approach, which models connections between words to add context and improve understanding of emotional nuance. Other studies used basic Machine Learning models, such as Suhasini's work using the Naive Bayes and KNN models for emotion detection in tweets. The Naive Bayes approach, which is based on applying Bayes' theorem and assumes that features are conditionally independent, predicts the most likely emotion by calculating the probability of each emotion given the words present in a tweet. Chiorrini builds on this using fine-tuned BERT models to classify emotions in tweets and achieved an accuracy of 92 percent, highlighting the advantages of deep contextual embeddings.

We focused the scope of our project on a set of primary emotions that are most individually distinguishable: joy, sadness, anger, fear, love, and surprise. In the training set, there are 5,362 tweets labeled as joy, 4,666 as sadness, 2,159 as anger, 1,937 as fear, 1,304 as love, and 572 as surprise. We train and compare several models to assess their ability in detecting these emotions. We expand on previous work by not only measuring model accuracy but also closely analyzing errors and attempting to address them. Understanding where and why the models make mistakes helps us identify the specific language cues and situations that still cause problems for automated emotion detection. By examining the model misclassifications, we provide a more complete view of what makes emotion recognition in text difficult and explore ways these errors can be reduced in future approaches.

4 - Models

4.1 - Baselines

Our primary objective is to classify the emotional tone of a message by assigning it to one of six categories: joy, sadness, anger, fear, love, or surprise, and evaluate if the prediction matches the label provided in the dataset. We measure model performance using the macro-averaged F1 score as it gives equal weight to each emotion regardless of how frequently it appears in the dataset. We prioritize this metric because the data is imbalanced; for example, “joy” has over 5,000 examples while ”surprise” has fewer than 600. Other researchers in this area, such as Saravia et al. (2018), also use accuracy and macro-F1 as evaluation metrics for emotion detection on social media data, allowing us to also directly compare results. As part of model training, we also apply class weights during training to each model to ensure that less common emotions are not overlooked.

To establish a baseline for evaluating more advanced models like BERT and CNN, we first train and assess a Multinomial Naive Bayes classifier using TF-IDF vectorization with up to 5,000 features to capture both single words and word pairs. This resulted in an accuracy of 82% and a macro-averaged F1 score of 67%. Based on these results, we set a target accuracy of at least 85% for the DAN, CNN, and BERT models, with the goal of achieving final performance in the mid-90% range.

4.2 - Deep Averaging Networks (DAN)

Deep Averaging Network (DAN) is a neural network architecture that represents a sentence by averaging word embeddings and then passing the result through multiple layers, making it a simple and efficient architecture commonly used in NLP tasks. DAN offers a way for the model to learn from the actual language in tweets instead of defaulting to the most common class. Our goal is not only to surpass the Naive Bayes baseline and DAN benchmark in overall accuracy, but also to understand where and why the models fail, especially with less common emotions like “surprise” and “fear.” A strong model should improve both overall results and recall for these underrepresented categories.

4.3 - Convolutional Neural Network (CNN)

To address the limitations of DAN, we implemented a Convolutional Neural Network (CNN) designed to capture more nuanced emotion-bearing patterns within text. The architecture uses multiple filter sizes in the convolutional layers, which allows the model to detect emotion related n-grams, such as short phrases or recurring word combinations. These local patterns often serve as strong indicators of sentiment.

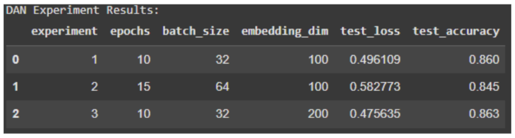
4.4 - Bidirectional Encoder Representations from Transformers (BERT)

Building on the improvements from the CNN, we incorporated more advanced transformer-based models, specifically Bidirectional Encoder Representations from Transformers (BERT). Unlike DAN and CNN, which use static or locally focused representations, these transformer models generate contextual embeddings for each word. This approach enables the model to capture subtle cues, such as shifts in tone or meaning that depend on context, which are often essential for accurately identifying emotions in text.

5 - Model Results

5.1 - DAN

For the DAN model, we performed 3 different experiments. The first experiment consisted of lower epochs, batch size, and embedding dimensions. The most optimal experiment yielded a test accuracy of 86% and an F1 score of 86%. Across all experiments, we noticed a significant amount of test loss, indicating that the model was generalizing patterns it learned from the training dataset. We concluded that the DAN model had the highest accuracy with a moderate number of epochs, smaller batch sizes, and higher embedding dimensions.



DAN Experiment Results:

	experiment	epochs	batch_size	embedding_dim	test_loss	test_accuracy
0	1	10	32	100	0.496109	0.860
1	2	15	64	100	0.582773	0.845
2	3	10	32	200	0.475635	0.863

Figure: DAN Results

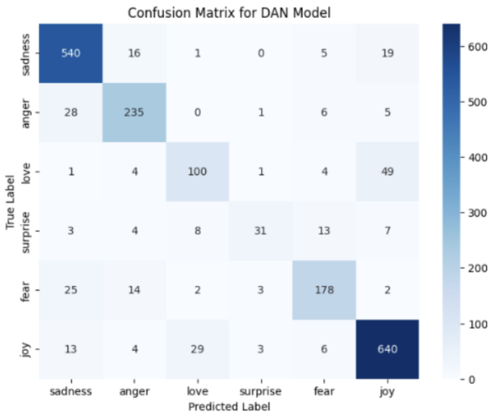
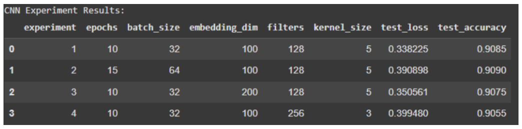


Figure: DAN Confusion Matrix

According to the confusion matrix, we also see that the DAN model commonly misclassifies sentiments of joy with love, sadness with anger, and sadness with fear.

5.2 - CNN

The CNN model achieved an accuracy of 91% and a macro F1 score of 88%, a substantial improvement over the DAN model. We ran a total of four experiments that consisted of changing the number of epochs, batch sizes, embedding dimensions, filters, and kernel sizes. The CNN model performed significantly better as the overall test loss decreased and the test accuracy increased across all experiments. We noticed that increasing the batch sizes would increase the test loss percentage, which signifies that the model is generalizing from the training dataset or overfitting. The model performed best with a moderate number of batch sizes, smaller batch sizes, lower embedding dimensions, lower filters, and greater kernel sizes.



CNN Experiment Results:

	experiment	epochs	batch_size	embedding_dim	filters	kernel_size	test_loss	test_accuracy
0	1	10	32	100	128	5	0.338225	0.9085
1	2	15	64	100	128	5	0.390898	0.9090
2	3	10	32	200	128	5	0.350561	0.9075
3	4	10	32	100	256	3	0.399480	0.9055

Figure 1: CNN Results

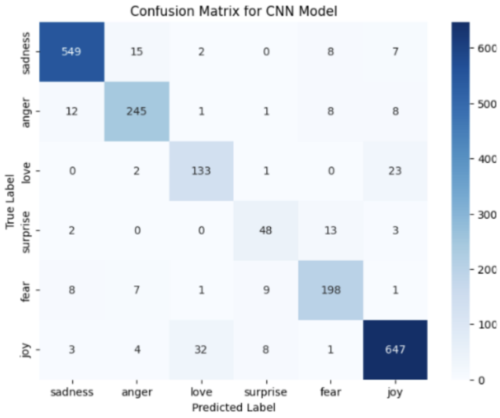
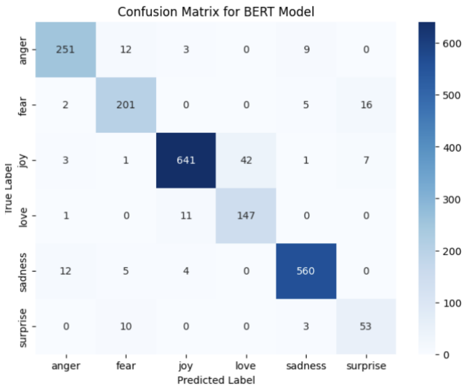


Figure 2: CNN Confusion Matrix

We can observe through the confusion matrix for the CNN model that the model did a much better job of correctly predicting the true labels compared to DAN. However, similar to DAN, we noted that the model more often mislabeled less distinct emotions. For example, positive emotions such as love and joy were confused with each other at a greater rate while emotions like joy and sadness were less likely to be mixed up. These results align closely with the findings reported by Abas et al. (2022), who observed similar confusion patterns in their BERT-CNN model, particularly among semantically related emotions like guilt and shame, or anger and fear. Despite differences in datasets and model complexity, both their study and our model show that even advanced neural architectures tend to misclassify emotions that share overlapping linguistic cues. This consistency suggests that such confusion is an inherent challenge in text-based emotion classification, highlighting an important area for future research and model development.

5.3 - BERT

The BERT model achieved the highest performance with a macro F1 score of 93%, beating the CNN's accuracy of 91%. Notably, it improved recall for the weakest classes, ”fear” and ”surprise”. We ran 3 different experiments for the BERT model. Specifically, we experimented with varying learning rates (Learning Rate: 1e-5, 5e-5, 1e-4 (Initial run used 2e-5)), varying number of epochs (Number of Epochs: 2, 4, 5 (Initial run used 3)), and varying batch sizes (Per Device Eval Batch Size: 8, 32 (Initial run used 16)). The changes between results were mostly immaterial, noting a typically +/- 1% difference to the test accuracy (93%) and F1-score (93%) throughout all the experiments. The BERT model's confusion matrix demonstrates a clear advance over the CNN and DAN models, particularly in accurately predicting joy and love. Compared to earlier results, BERT reduced misclassifications among positive emotions and improved precision across most classes. While some confusion remains, such as sadness occasionally being labeled as anger or fear as surprise, BERT's overall performance highlights its superior ability to capture the emotional nuances in tweets compared to simpler neural architectures.



	anger	fear	joy	love	sadness	surprise
anger	251	12	3	0	9	0
fear	2	201	0	0	5	16
joy	3	1	641	42	1	7
love	1	0	11	147	0	0
sadness	12	5	4	0	560	0
surprise	0	10	0	0	3	53

Figure 3: BERT Confusion Matrix

6 - Analysis of Errors

6.1 - Common Errors

Error Pairs: Across all models, the most common mistakes occurred between emotions that are close in meaning, such as joy and love, sadness and fear, or fear and anger. Even BERT occasionally confused “surprise” with “joy,” likely due to the presence of similar words in the tweets, such as “wow” or “unbelievable.” In the example above, BERT predicted “surprise” when the true label was “fear,” which may be attributed to the model not recognizing that metformin is a medication. Had the model identified this context, it might have inferred that the person felt “weird” due to anxiety, and thus fear, before taking the medication. These errors demonstrate that even the best models can still struggle with missing background knowledge or subtle context that people naturally pick up on.

Data Noise: Manual inspection of mislabeled tweets revealed inconsistencies in the dataset. Some tweets labeled as “joy” actually conveyed low-energy emotions, which were more indicative of “melancholy” or “neutrality.” This type of label noise likely limited the maximum achievable performance of our models. For example, the tweet “I feel a little mellow today;joy” would be more accurately classified as sadness.

Lack of Context: Some sentences in the dataset were missing special characters or lacked surrounding context, which often led to confusion for the models. The dataset was preprocessed to remove extra characters like &, !, or other symbols that might interfere with parsing. Although this improved consistency, it also eliminated punctuation such as exclamation marks or question marks from the ends of sentences, which are important cues for both humans and models to understand emotion. Additionally, the lack of context, with sentences presented without the preceding or following text, made it challenging to accurately interpret the intended emotion.

Class Imbalances: Class imbalance also contributed to the errors we observed. As described in the Dataset Overview, some emotion classes contained over 2,000 examples, while others had fewer than 1,000. This uneven distribution led to a bias in model training, with larger classes exerting more influence on the results. To address this issue, we had already structured our experiments around the macro-averaged F1 metric and applied class weights during training to ensure that less represented emotions received appropriate attention.

6.2 - Tackling the Errors

Error patterns are particularly challenging because the dataset contains varying intensities of each emotion, as well as slang and sarcasm, which models often struggle to detect. As a result, some phrases could plausibly be classified into multiple emotion categories, even though only one label is allowed. To address this, we developed a final ensemble model that assigns each test sentence the label predicted by the majority of all models. This approach works because each model, due to its unique architecture and parameters, picks up on different cues within the text. By aggregating predictions from DAN, CNN, BERT, and the baseline model, the ensemble produced an output that outperformed any individual model, achieving an accuracy of 95

For data noise, there was little we could do since the errors were present in the original dataset. However, after randomly sampling ten times with a sample size of 50, we found misclassifications to be rare. Typically, we observed only 1 or 2 cases every three runs (about 1%). Given the overall dataset size of 20,000 sentences, these rare occurrences had minimal impact and were effectively filtered out by the models.

For lack of context, as with data noise, we are not able to faithfully restore the missing punctuation and context. To assess the impact, we conducted random sampling ten times with a sample size of 50. Misclassifications due to missing context were rare - only one case per run on average (about 2%). While this was slightly higher than for data noise, the effect was still small enough to be considered negligible.

7 - Conclusion

This project demonstrated the effectiveness of deep learning models for emotion recognition in informal online texts, particularly tweets. Our experiments showed that while simple models like DAN performed reasonably well, more advanced architectures like CNNs and transformer-based models such as BERT offered significant improvements in both accuracy and recall. Each model was able to capture different aspects of emotional nuance in text. By combining these models in an ensemble, we achieved results that exceeded the performance of any individual model, achieving 95% accuracy and a macro F1-score well above baseline expectations, which is particularly important given the class imbalance. These results highlight the advantages of ensemble learning when working with subjectively labeled, high-variance data such as emotional language in tweets. The errors we observed point to several directions for future work, including improving classification for underrepresented emotions like “surprise,” incorporating methods for detecting sarcasm, and expanding the label set to account for compound or multi-label emotional states. Future iterations of this project could integrate additional context-aware models, such as attention-based transformers or large-scale LLM embeddings, to improve handling of subtle tone shifts and ambiguous sentiment. Additionally, more robust preprocessing pipelines and semi-supervised approaches may help mitigate the impact of label noise in the dataset. Our work offers a starting point for building more trustworthy emotion recognition and shows that using a variety of model approaches helps us better capture the complexity of human language.

8 - Project Contributions

In terms of our project contributions, each of us attempted to work on different models. Suvass focused on the Baselines and helped build out the Ensemble model that we developed. Augustin worked on DAN and CNN and worked on getting them up to a good accuracy. Tommy was able to work on BERT and hyperparameter-tune BERT to get great results. In terms of the paper, each individual focused on writing up their own model and analysis of their results. Tommy worked on the Abstract, Introduction and Data Overview. Suvass worked on the Error Analysis and the Conclusion. Augustin worked on his DAN and CNN sections as well as adding on to other parts of the paper. Although each of us were assigned certain models we worked as a group on each of them making our contributions 33% each.

9 - References

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized Affect Representations for Emotion Recognition. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics. Bharti, Santosh Kumar et al. "Text-Based Emotion Recognition Using Deep Learning Approach." Computational intelligence and neuroscience vol. 2022 2645381. 23 Aug. 2022, doi:10.1155/2022/2645381

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

A. Chiorrini, C. Diamantini, A. Mircoli and D. Potena, "Emotion and sentiment analysis of tweets using BERT," in EDBT/ICDT Workshops, Nicosia, Cyprus, 2021

A. R. Abas, I. Elhenawy, M. Zidan, and M. Othman, "BERT-CNN: A Deep Learning Model for Detecting Emotions from Text," Comput. Mater. Contin., vol. 71, no. 2, pp. 2943–2961, 2022. https://doi.org/10.32604/cmc.2022.021671