Emotion Classification: Cross-Dataset Performance Analysis

Dogăreci Bianca-Alexandra
biancadogareci@ymail.com

Potlog Ioana
ioanapotlog@gmail.com

Negoiţă-Crețu Raluca-Marina
ralucanegoita13@gmail.com

Spataru Mara-Andreea
mara.spataru03@gmail.com

Abstract

This analysis examines the performance of text-based emotion classification models across different datasets.

1 Introduction

These experiments were conducted:

1. **In-Domain Evaluation**: Training and testing on Dataset 1 (10 emotions) / Dataset 2 (6 emotions);

2. **Cross-Domain Generalization (+ Filtered Evaluation)**: Training on Dataset 1 and testing on Dataset 2 and training on Dataset 2 and testing on Dataset 1.

The results reveal significant challenges in cross-dataset emotion classification, with models performing substantially better when tested on data similar to their training set. The analysis provides insights into emotion classification generalization, vocabulary differences between datasets and class imbalance effects.

2 State of the art - SOTA

Emotion Analysis (EA) differs fundamentally from sentiment analysis. While sentiment analysis focuses on identifying positive, negative, or neutral attitudes, EA seeks to classify specific emotions such as anger, joy, fear, sadness, disgust, and surprise (Plaza-del-Arco et al., 2024). Common EA tasks include:

- Emotion recognition/classification (identifying expressed emotions),

- Emotion cause detection (understanding triggers behind emotions),

- Emotion intensity prediction (measuring emotional strength).

3 Proposed Implementation

3.1 Selected Methods

Inspired by the reviewed literature, we propose to implement several classical machine learning models, consistent with prior successful approaches:

- Traditional ML models:

1. Multinomial Naive Bayes;
2. Random Forest Classifier;
3. Logistic Regression;
4. Support Vector Machine (SVM);

- Deep Learning models: Transformer-based models such as BERT.

Traditional machine learning models remain popular due to lower computational costs and ease of implementation, whereas deep learning models, although more resource-intensive, often outperform them in handling complex and nuanced emotional content.

3.2 Dataset Overview

The experiments involved two distinct emotion-labeled text datasets:

- Dataset 1: Contains 10 emotion categories (anger, anticipation, disgust, fear, joy, love, neutral, sadness, surprise, trust);

- Dataset 2: Contains 6 emotion categories (anger, fear, joy, love, sadness, surprise);

The difference in the number and type of emotion categories can lead to label mismatch or loss of information. For example, if the model is trained on Dataset 1, it learns to recognize and distinguish all 10 emotions. However, when it is tested on Dataset 2, there is no data corresponding to 4 of the emotions it was trained on, so the model's learned representations for those categories become irrelevant or even misleading during evaluation.

In addition, if a model is trained on Dataset 2 and then tested on Dataset 1, it has never seen examples of some emotions during training. As a result, the model is forced to misclassify those unseen categories as one of the six it does know.

The datasets were obtained from publicly available Hugging Face repositories. Each consists

of short, emotion-labeled texts (content;label). We manually added 300 additional AI generated samples to Dataset 1 for the "love" category that existed in Dataset 2, but not in Dataset 1. Emotion distribution varies, with some classes more frequent than others as we can see in the histograms below.

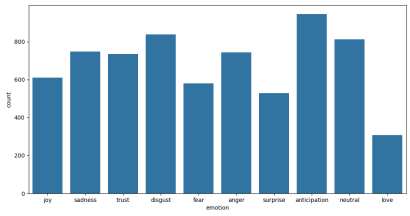

Figure 1: Frequency of Emotion Categories in Dataset 1

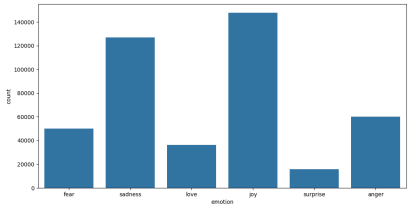

Figure 2: Frequency of Emotion Categories in Dataset 2

3.3 Preprocessing data

Text preprocessing is a crucial step in order to improve the performance of emotion classification models. The following steps were applied:

- **Removing punctuation** as they do not contribute meaningful information to emotion detection and replacing emojis with suggestive words (":(" -> sad);

- **Tokenization** converts a sentence into a list of words, which is essential for further steps;

- **Stop word removal** - stop words ("a", "the", "is", "are") are common words in English that carry minimal semantic value. These were removed to reduce noise in the data;

- **Lemmatization** reduces words to their root form by considering their morphological structure ("tickets" -> "ticket").

These preprocessing steps were chosen to reduce noise, standardize word forms, and retain only the most meaningful information for accurate emotion classification.

To better understand the structure of the datasets, we analyzed the number of words remaining after lemmatization. Most sentences are short, as shown in the histograms below.

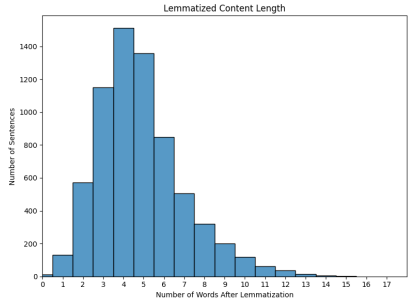

Figure 3: Sentence Length Distribution After Lemmatization – Dataset 1

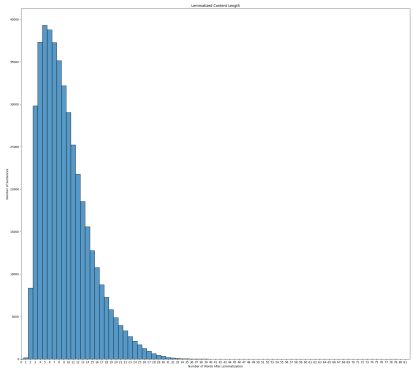

Figure 4: Sentence Length Distribution After Lemmatization – Dataset 2

3.4 Feature Extraction

After text pre-processing, all text input must be converted into numerical form (well-defined, fixed-length numerical representations) before it can be used by a machine learning model.

In supervised learning, two widely used techniques for feature extraction from text are:

- **Count Occurrence** - counts how many times each word appears in a sentence, representing text as a vector of word frequencies;

- **TF-IDF** - adjusts word frequency by how common the word is across all documents, giving more weight to informative and distinctive words.

However, both methods produce *sparse* and *context-independent* representations. This means they do not preserve word meaning or capture the surrounding context in which a word appears, which means some predictions may be inaccurate later.

3.5 Comparative Analysis of Methods

Peng et al. (2022) find that traditional ML models provide efficient baselines, while deep learning models offer superior performance but at the cost of higher complexity and computational demands.

Plaza-del-Arco et al. (2024) further emphasize that even the best-performing models often ignore crucial demographic factors such as age, gender, and culture, which significantly affect emotional expression and perception.

3.6 Method - Multinomial Naive Bayes

The Multinomial Naive Bayes classifier is a probabilistic machine learning algorithm based on Bayes' theorem with an assumption of conditional independence between features. It is particularly well-suited for text classification tasks, where documents are represented as word frequency vectors. The algorithm calculates the probability of a text belonging to each emotion class based on the frequency of words appearing in that text, and then assigns the most probable class as the prediction.

Unlike other models, Multinomial Naive Bayes assumes that all features (words) are independent of each other given the class label, which simplifies computation but ignores word relationships. Despite this simplifying assumption, it often performs surprisingly well for text classification tasks, offering a good balance between computational efficiency and predictive performance.

3.6.1 Experiment 1: Model Trained and Tested on Dataset 1

Results:

- Accuracy: 85.01%
- Precision: 86.86%
- Recall: 81.84%
- F1-Score: 83.13%

Performance by Emotion Class:

- Strongest performance: love - perfect precision (1.00) but lower recall (0.49) - the model never makes false positive predictions for love but misses many instances;

- Other strong performers: disgust (F1: 0.92), trust (F1: 0.91), and anticipation (F1: 0.86) - consistently high precision and recall;

- Weakest performance: love - despite perfect precision, its low recall (0.49) results in a comparatively weak F1-score (0.66).

Confusion Matrix Analysis:

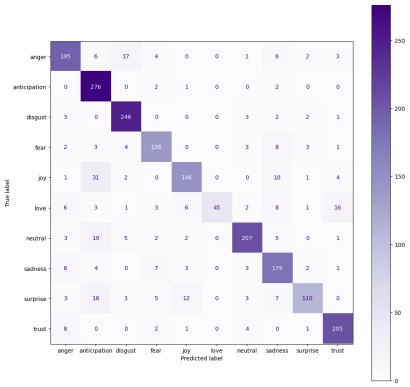

Figure 5: Confusion Matrix

The confusion matrix reveals several notable patterns:

- Most emotions show strong diagonal values, indicating good classification accuracy.

- Joy is sometimes misclassified as anticipation (31 instances), with smaller numbers misclassified as sadness (10) and trust (4).

- Love, despite its perfect precision, is frequently misclassified as trust (16 instances), sadness (8), anger (6), and joy (6), explaining its lower recall.

- Surprise shows confusion with anticipation (18 instances) and joy (12 instances), reflecting the challenge in distinguishing between these emotions.

3.6.2 Experiment 2: Model Trained on Dataset 1, Tested on Dataset 2

Results:

- Accuracy: 20.67%
- Precision: 19.46%
- Recall: 8.45%
- F1-Score: 8.72%

Performance by Emotion Class:

The model shows extremely poor generalization to Dataset 2, with multiple emotion categories (anticipation, disgust, neutral, trust) completely unrecognized as they do not exist in the test set. Among emotions present in Dataset 2:

- Sadness is the best-recognized emotion with an F1-score of 0.42 and the highest recall (0.57), indicating the model frequently defaults to this prediction.

- Joy and fear show very poor performance despite being present in both datasets (F1-scores of 0.09 and 0.16 respectively).

- Love and surprise are barely recognized at all (F1-scores below 0.07).

Confusion Matrix Analysis:

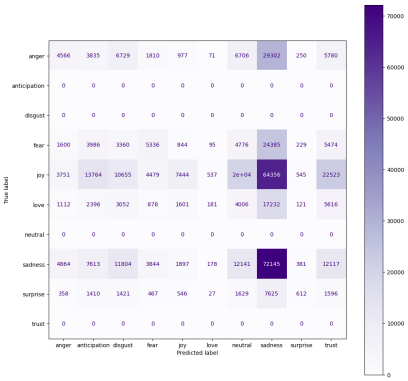

Figure 6: Confusion Matrix

The confusion matrix reveals:

- A strong bias toward predicting sadness, with the model assigning this label to many samples regardless of their true emotion, with sadness being the most frequently predicted class by a significant margin.

- The second most predicted emotion is joy, although at much lower rates than sadness.

- Very few correct classifications across most emotion categories, with many emotions being almost entirely misclassified.

- Systematic misclassification patterns that suggest fundamental differences in how emotions are expressed between the datasets.

3.6.3 Experiment 3: Model Trained on Dataset 1, Tested only on the present emotions in Dataset 2

Results:

- Accuracy: 20.67% (unchanged, as expected)
- Precision: 32.43%
- Recall: 14.09%
- F1-Score: 14.53%

Performance by Emotion Class:

When evaluating only the emotions present in Dataset 2, the model still shows poor generalization, though metrics improve somewhat when irrelevant emotions are excluded from calculation:

- Sadness remains the best recognized emotion (F1: 0.42), with both reasonable precision (0.34) and the highest recall (0.57).

- Joy has the highest precision (0.56) but very low recall (0.05), indicating the model rarely predicts this class but is often correct when it does.

- Fear and anger show marginally better performance (F1: 0.16 and 0.12 respectively) but remain poorly classified.

- Love performs extremely poorly (F1: 0.01), showing almost no successful classifications despite being present in both datasets.

Confusion Matrix Analysis:

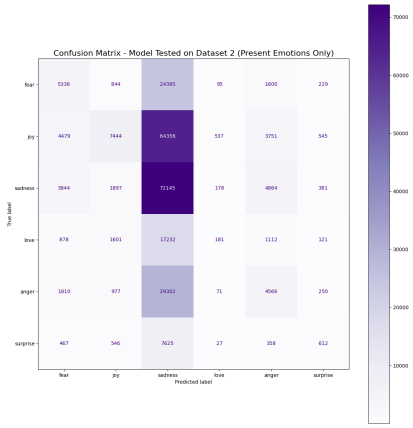

Figure 7: Confusion Matrix

The filtered confusion matrix focusing only on emotions present in both datasets shows:

- An overwhelming bias toward sadness predictions, which dominates the model's output. The visualization clearly shows sadness is predicted far more frequently than any other emotion.

- Extremely poor classification performance for love and surprise, with love having only 181 correct predictions out of 36,195 instances.

- While joy has some correct predictions, a vast majority of joy instances (64,356) are misclassified as sadness, highlighting the model's strong bias toward this emotion class and the substantial differences in how emotions are expressed between datasets.

3.7 Method - Random Forest Classifier

The Random Forest Classifier works by building multiple decision trees on random subsets of the data, and then combining their predictions. Each tree votes, and the category with the majority vote becomes the final prediction.

3.7.1 Experiment 1: Model Trained and Tested on Dataset 1

Results:

- Accuracy: 71.87%
- Precision: 71.97%
- Recall: 70.17%
- F1-Score: 70.63%

Performance by Emotion Class:

- Strongest performance: disgust - very high precision (0.84) and recall (0.83) - consistent performance;

- Weakest performance: love - very low recall (0.51) - the model misses a lot of "love" examples;

Confusion Matrix Analysis:

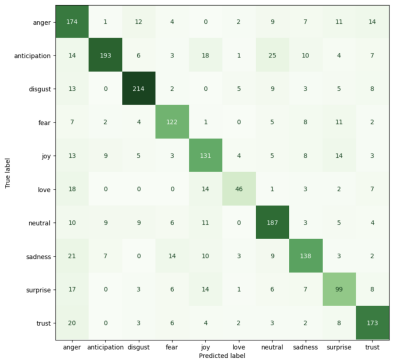

Figure 8: Confusion Matrix

3.7.2 Experiment 2: Model Trained on Dataset 1, Tested on Dataset 2

Results:

- Accuracy: 11.54%
- Precision: 13.75%
- Recall: 8.51%
- F1-Score: 7.68%

Performance by Emotion Class:

The model's overall performance is very poor, and no class stands out as strong. Multiple classes, including anticipation, disgust, neutral, and trust, have 0.00 F1-score, indicating the model completely failed to recognize these emotions.

Confusion Matrix Analysis:

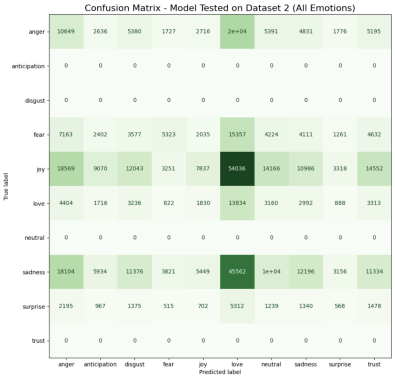

Figure 9: Confusion Matrix

3.7.3 Experiment 3: Model Trained on Dataset 1, Tested only on the present emotions in Dataset 2

Results:

- Accuracy: 11.54%
- Precision: 22.9%
- Recall: 14.18%
- F1-Score: 12.81%

Performance by Emotion Class:

When evaluating only the present emotions in Dataset 2, the classifier continues to show weak generalization. The highest F1-scores are observed for anger (0.18) and fear (0.16), but these are still far from acceptable performance. Common emotions like joy (F1: 0.09) and surprise (F1: 0.04) are particularly poorly classified.

Confusion Matrix Analysis:

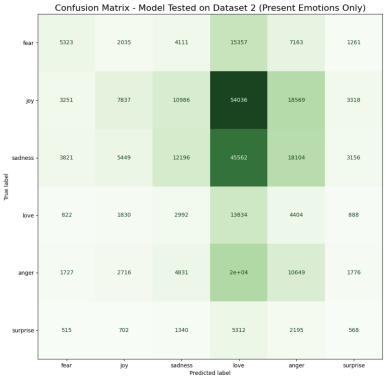

Figure 10: Confusion Matrix

3.8 Method - Logistic Regression

The Logistic Regression Classifier works by assigning weights to each input feature (such as words in a sentence) and calculating a score for each emotion class. These scores are then converted into probabilities using a softmax function. The emotion with the highest probability is selected as the final prediction.

3.8.1 Experiment 1: Model Trained and Tested on Dataset 1

Results:

- Accuracy: 88.46%
- Precision: 88.08%
- Recall: 86.97%
- F1-Score: 87.42%

Performance by Emotion Class:

- Strongest performance: disgust - very high precision (0.96) and recall (0.93) - consistent performance;

- Weakest performance: surprise - although precision is acceptable, the lower recall (0.76) indicates the model misses a noticeable number of surprise examples, making it the weakest class in terms of overall F1-score (0.79)

Confusion Matrix Analysis:

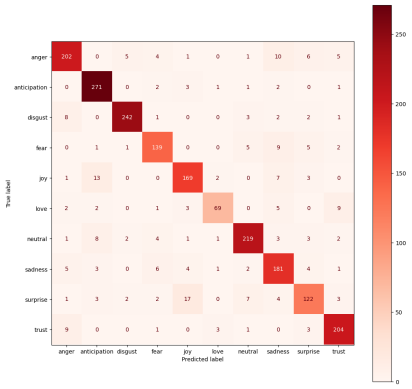

Figure 11: Confusion Matrix

3.8.2 Experiment 2: Model Trained on Dataset 1, Tested on Dataset 2

Results:

- Accuracy: 20.56%
- Precision: 19.44%
- Recall: 9.03%
- F1-Score: 9.40%

Performance by Emotion Class:

The model performs extremely poorly on Dataset 2, with an overall F1-score of only 9.4% and multiple emotion categories such as anticipation, disgust, neutral, and trust completely unrecognized (F1-score of 0.00), highlighting a severe failure to generalize across datasets.

Confusion Matrix Analysis:

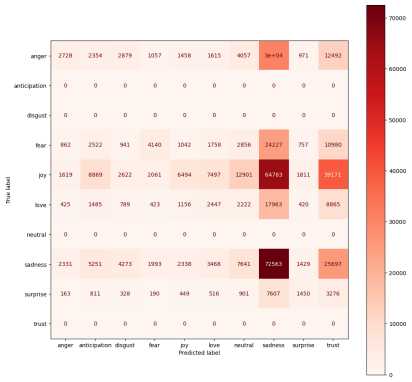

Figure 12: Confusion Matrix

3.8.3 Experiment 3: Model Trained on Dataset 1, Tested only on the present emotions in Dataset 2

Results:

- Accuracy: 20.56%
- Precision: 32.40%
- Recall: 15.05%
- F1-Score: 15.67%

Performance by Emotion Class:

When evaluating only the present emotions in Dataset 2, the classifier continues to show weak generalization. The highest F1-scores are observed for sadness (0.42) and fear (0.14), but these are still far from acceptable performance. Common emotions like joy (F1: 0.08) and anger (F1: 0.08) are particularly poorly classified.

Confusion Matrix Analysis:

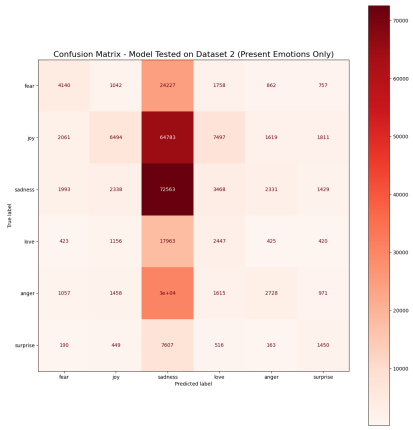

Figure 13: Confusion Matrix

3.9 Method - Support Vector Machine

The Support Vector Machine (SVM) classifier works by finding the optimal hyperplane that separates data points belonging to different emotion classes. Each input text is transformed into a high-dimensional vector representation (typically using TF-IDF features), and the SVM algorithm identifies the decision boundaries that maximize the margin between emotion classes. Unlike probabilistic models, SVM directly focuses on classification by learning from support vectors — the critical data points closest to the margin. The model assigns a class label based on which side of the hyperplane the input vector falls on.

3.9.1 Experiment 1: Model Trained and Tested on Dataset 1

Results:

- Accuracy: 89.44%

- Precision: 88.31%

- Recall: 88.22%

- F1-Score: 88.25%

Performance by Emotion Class:

- Strongest performance: disgust - very high precision (0.95) and recall (0.95) - consistent performance;

- Weakest performance: surprise - although precision is acceptable, the lower recall (0.77) indicates the model misses a noticeable number of surprise examples, making it the weakest class in terms of overall F1-score (0.78)

Confusion Matrix:

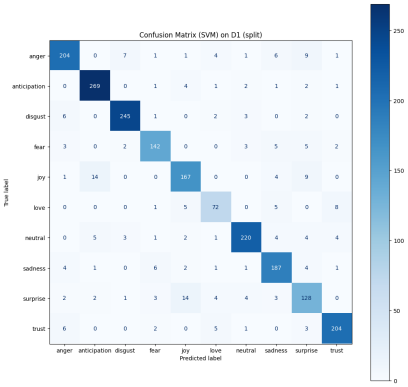

Figure 14: Confusion Matrix

3.9.2 Experiment 2: Model Trained on Dataset 1, Tested on Dataset 2

Results:

- Accuracy: 19.15%

- Precision: 19.93%

- Recall: 9.71%

- F1-Score: 11.16%

Performance by Emotion Class:

The model shows very weak performance on Dataset 2, achieving an overall F1-score of just 11.16%. Several emotion classes—such as anticipation, disgust, neutral, and trust—are not detected at all (F1-score of 0.00), indicating a significant inability to generalize to new data.

Confusion Matrix:

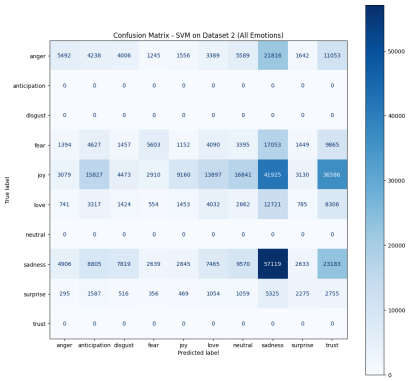

Figure 15: Confusion Matrix

3.9.3 Experiment 3: Model Trained on Dataset 1, Tested only on the present emotions in Dataset 2

Results:

- Accuracy: 19.15%
- Precision: 33.21%
- Recall: 16.19%
- F1-Score: 18.60%

Performance by Emotion Class:

When evaluating only the present emotions in Dataset 2, the classifier continues to show weak generalization. The highest F1-scores are observed for sadness (0.40) and fear (0.18), but these are still far from acceptable performance. Common emotions like joy (F1: 0.11) and anger (F1: 0.14) are particularly poorly classified.

Confusion Matrix:

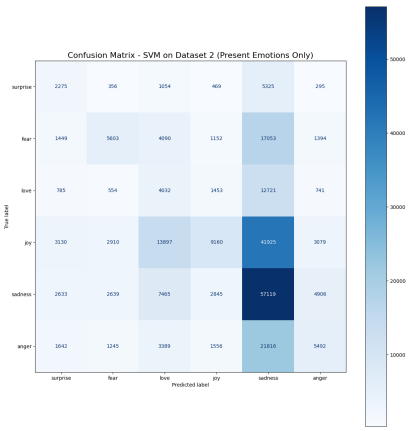

Figure 16: Confusion Matrix

3.9.4 Contextual Feature Extraction

To improve performance and address the limitations of traditional feature extraction methods, we changed our approach by using BERT, a more advanced and context-aware model. Instead of Count or TF-IDF vectors, BERT uses

tokenization and pre-trained embeddings to convert text into meaningful numerical representations, capturing the context and semantics of words. This allowed us to test whether a deeper, transformer-based model could provide better results, especially in cross-dataset evaluation.

3.10 Method - BERT

BERT is a powerful deep learning model developed by Google that understands language by reading it in both directions—left to right and right to left. This helps it grasp the context of each word more accurately than traditional models.

Before BERT can process text, it breaks sentences into pieces called tokens. These tokens are then converted into numbers (token IDs) so the model can work with them. Words are sometimes split into subwords if they aren't in BERT's vocabulary.

A special token called [CLS] is added at the start of each sentence. After processing, this token holds a summary of the input and is used to decide the final emotion label.

Unlike classic models that rely on manually selected features, BERT learns directly from the data. This makes it better at picking up subtle language patterns and meanings.

3.10.1 Setup: Fine-Tuning BERT on Dataset 1

BERT was fine-tuned on Dataset 1 using the HuggingFace Trainer class. The model used is bert-base-uncased with a classification head adapted to the number of emotion classes. The training was performed for 3 epochs with a batch size of 16, evaluation at every 10 steps and weight decay for regularization.

Custom metrics such as accuracy, precision, recall and F1-score were defined to evaluate the model. The dataset was tokenized with a maximum length of 128 tokens and inputs were padded/truncated accordingly.

To better understand training behavior, two key graphs were generated and included:

Training Loss Curve: Shows how the model's error decreases over time.

Epoch Progression: Indicates how steps progress throughout the training epochs.

These plots offer visual confirmation that the model successfully converged during training.

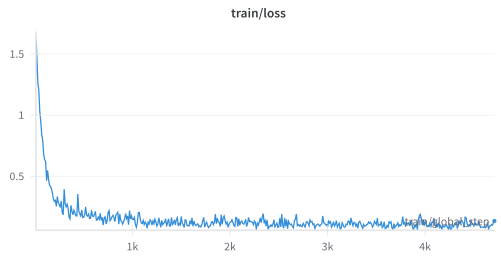

Figure 17: BERT Dataset 1 - Training Loss Curve

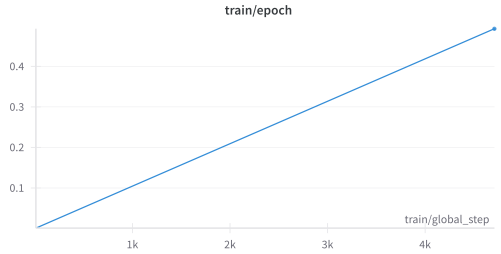

Figure 18: BERT Dataset 1 - Epoch Progression

3.10.2 Setup: Fine-Tuning BERT on Dataset 2

BERT was fine-tuned on Dataset 2 using the HuggingFace Trainer class. The model used is bert-base-uncased, with the classification head adjusted to match the number of emotion classes. Due to runtime constraints, the training was limited to 1 epoch, with a batch size of 32 and evaluation performed every 500 steps. Weight decay and mixed-precision (fp16) were also used to optimize training.

Evaluation metrics included accuracy, precision, recall, and F1-score, computed using macro averaging. The text inputs were tokenized with a max length of 128 tokens and appropriately padded/truncated.

To support training diagnostics, two graphs are provided:

Training Loss Curve: Illustrates how the model's loss evolved over time.

Epoch Progression: Tracks how the model advanced through steps within the epoch.

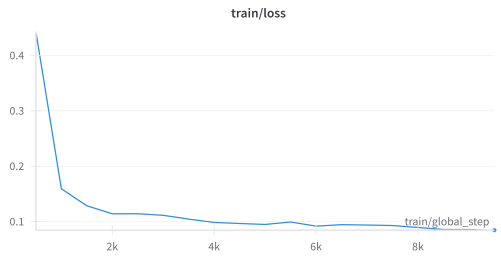

Figure 19: BERT Dataset 2 - Training Loss Curve

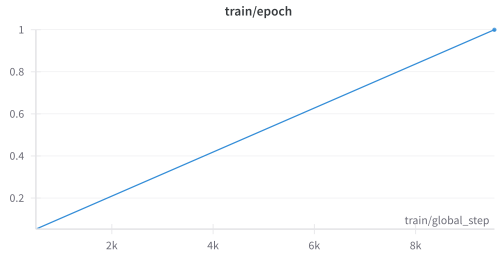

Figure 20: BERT Dataset 2 - Epoch Progression

3.10.3 Experiment 1: Model Trained and Tested on Dataset 1

Results:

- Accuracy: 96.39%
- Precision: 96.41%
- Recall: 96.11%
- F1-Score: 96.23%

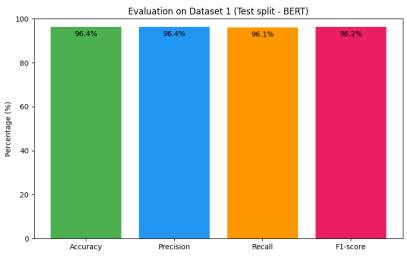

Figure 21: BERT Evaluation Metrics on Dataset 1

Performance by Emotion Class:

Unlike traditional models like SVM or Logistic Regression, BERT delivers exceptionally consistent results, with all emotion classes scoring F1-scores above 0.90.

- Top performer: love – Perfect precision (1.00) and high recall (0.96), showing the model never misclassifies this emotion.

- Most stable: trust – Achieves 100% recall and 0.98 precision, showing excellent balance.

- Lowest (but strong): surprise – With an F1-score of 0.92, it's the weakest among the group, mainly due to a slightly lower recall (0.90), yet still highly reliable.

These results emphasize BERT's strong ability to understand emotional nuance, far beyond what manual methods can typically achieve.

Normalized Confusion Matrix:

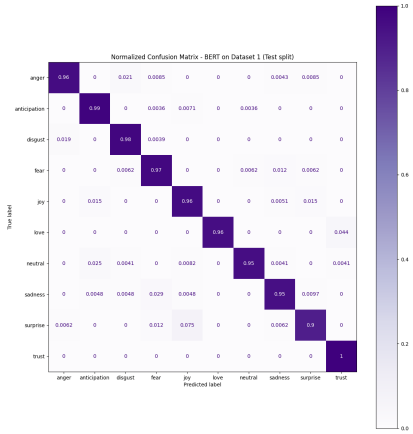

Figure 22: Normalized Confusion Matrix

3.10.4 Experiment 2: Model Trained on Dataset 1, Tested on Dataset 2

This experiment evaluates how well a BERT model trained on Dataset 1 generalizes to new data. Although the test is performed on a sample of 20,000 entries from Dataset 2 (to manage computation), results still reveal clear limitations in cross-domain generalization.

Results:

- Accuracy: 33.80%
- Precision: 48.95%
- Recall: 27.90%
- F1-Score: 33.84%

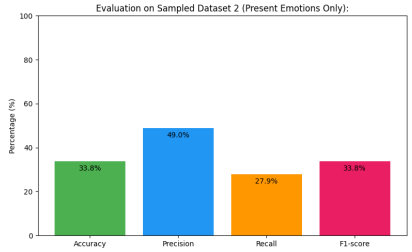

Figure 23: Trained Dataset 1 - BERT Evaluation Metrics on Dataset 2

Performance by Emotion Class:

While traditional models like SVM and Logistic Regression struggled heavily on this task (e.g., SVM: F1-score approximative 11%), BERT more than doubled performance, reaching a macro F1-score of 33.84%. Despite this, results still

indicate that the model fails to generalize reliably to Dataset 2.

This highlights that while BERT shows clear improvement over classical models, its cross-domain robustness still has major limitations, especially without fine-tuning on target-domain data.

Normalized Confusion Matrix:

Figure 24: Normalized Confusion Matrix

3.10.5 Experiment 3: Model Trained on Dataset 2, Tested on Dataset 2 and Dataset 1

Results tested on Dataset 2 split:

- Accuracy: 94.15%
- Precision: 93.52%
- Recall: 88.27%
- F1-Score: 90.23%

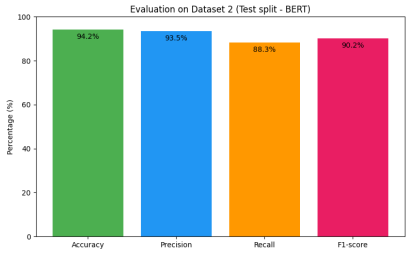

Figure 25: BERT Evaluation Metrics on Dataset 2

Performance by Emotion Class:

BERT demonstrates strong performance across all emotion classes, achieving high overall accuracy and a macro F1-score of 90

- Joy and Sadness are classified with high reliability (F1: 0.96–0.98).
- Fear is handled well, with high recall and F1 of 0.91.
- Love stands out with perfect precision (1.00), though lower recall suggests missed instances.
- Surprise remains the most challenging, with the lowest F1 (0.80).

This represents a significant improvement over traditional models, effectively doubling classification performance.

Normalized Confusion Matrix:

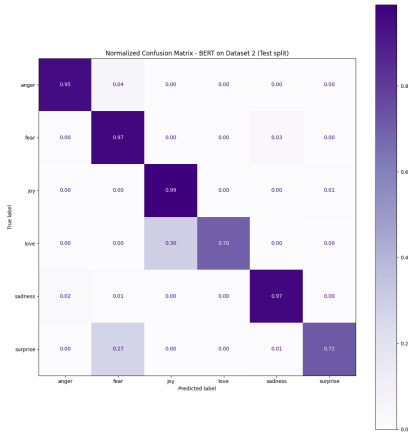

Figure 26: Normalized Confusion Matrix

Results testing on Dataset 1:

- Accuracy: 61.61%
- Precision: 58.91%
- Recall: 54.34%
- F1-Score: 48.59%

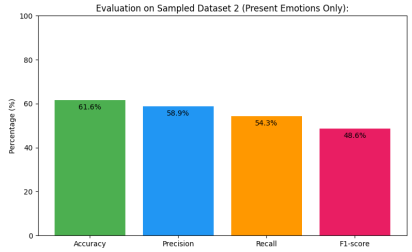

Figure 27: Trained Dataset 2 - BERT Evaluation Metrics on Dataset 1

Performance by Emotion Class:

Compared to traditional models like SVM, which performed poorly under cross-dataset evaluation (e.g., F1 - 26%), BERT exhibits significantly improved generalization. Emotions such as joy (94%), sadness (76%), and anger (75%) are recognized with high accuracy. Even fear shows solid performance (75%), although love and surprise remain difficult, often being misclassified as joy or anger. Despite these weaknesses, BERT more than doubles the overall performance of traditional classifiers, confirming its robustness in low-transfer settings.

Normalized Confusion Matrix:

Figure 28: Normalized Confusion Matrix

4 Future Work

We plan to continue improving this project by training additional models and training/testing across more diverse emotion datasets. This work could serve as a foundation for developing emotion-aware applications, such as chatbots, mental health tools or content recommendation systems.

5 Conclusion

Through this project, we learned how to apply various machine learning algorithms, both classical models and deep learning techniques like BERT. We gained practical experience with text preprocessing, feature extraction and model evaluation using accuracy, precision, recall and F1-score.

We especially enjoyed exploring how different models behave across datasets and understanding the challenges of cross-domain generalization. What we found most exciting was working with BERT and seeing how context-aware embeddings can improve performance.

On the other hand, we found the time-consuming nature of fine-tuning large models and dealing with hardware limitations (like needing GPU for BERT) to be frustrating at times.

Limitations

One of the primary limitations of this work is the computational cost associated with using transformer-based models such as BERT, which require significant GPU resources for training. This can limit scalability and accessibility, especially for users without access to powerful hardware. In contrast, traditional machine learning models such as Support Vector Machines (SVM), Multinomial Naive Bayes, Random Forest and Logistic Regression are much more lightweight and can be trained efficiently on CPUs.

All experiments were conducted in Google Colab, which by default uses CPU for training classical machine learning models. However, for fine-tuning with BERT, we switched the runtime environment to use a GPU, as transformer-based models require substantially more computational power.

Ethical Statement

Emotion detection can be misused for profiling, manipulation or surveillance without consent. Our datasets may include biases from social media, such as cultural or linguistic imbalances. We did not apply advanced debiasing methods. We recommend using this research responsibly. Emotion AI should only be applied with user awareness and ethical intent.

References

Cao L. Zhou Y. Ouyang Z. Yang A. Li X. Jia W. Yu S. Peng, S. 2022. A survey on deep learning for textual emotion analysis in social networks.

Curry A. Cercas Curry A. Hovy D. Plaza-del Arco, F. M. 2024. Emotion analysis in nlp: Trends, gaps and roadmap for future directions.