

Emotion Detection in Text Using a Lightweight Transformer Model

Presented by:
Chaturya Papajjagari
[My Github](#)

Introduction and Motivation

Objective:

- Build an efficient emotion detection system.
- Classify text into key emotions: Joy, Sadness, Love, Anger, Fear, and Surprise.
- Utilize a pre-trained model for rapid deployment without heavy training.

Why Emotion Detection Matters:

- Enhances customer service via sentiment analysis.
- Supports mental health initiatives.
- Improves chatbot and virtual assistant empathy.
- Analyzes social media trends and user behavior.

Model Selection

Chosen Model:

BERT - A smaller, faster, and nearly as accurate version of BERT(Bidirectional Encoder Representations from Transformers).

Why BERT?

- Reduces computational load.
- Maintains high accuracy.
- Ideal for lightweight applications.

Pretrained Model Used:

nateraw/bert-base-uncased-emotion

Dataset Overview

Dataset Used:

I have used the **Emotion Detection Dataset** (often known from sources like "Emotion Dataset" or "GoEmotions" from Google).

Dataset link: [Hugging face](#)

Dataset Details:

- **Total Examples:** Approximately **58,000** examples (depends if you are using the full version like GoEmotions or a smaller subset).
- **Number of Classes:** Typically **6 to 28** classes (depending on the version you use — we focus on major ones like Love, Joy, Sadness, etc.).
- **Type of Data:** Short sentences or text snippets labeled with corresponding emotions.
- **Data Split:**
 - **Training Set:** ~80% of data for model training.
 - **Validation Set:** ~10% for tuning hyperparameters.
 - **Test Set:** ~10% for final evaluation.

Methodology

Approach:

- Install necessary libraries (transformers, torch).
- Load the pre-trained model and tokenizer.
- Implement a smart prediction function.
- Apply thresholding to boost "Love" detection.
- Test with real-world input examples.

Platform:

I have used Google Colab as my platform to develop my project.

Example Predictions

Text	Predicted Emotion
"I love spending time with my family."	Love
"I'm scared about tomorrow's presentation."	Fear
"Today is an amazing day!"	Joy
"I am broken and sad."	Sadness

Challenges and Solutions

Challenges:

- Imbalanced dataset (few "Love" examples).
- Risk of biased predictions.

Solutions:

- Smart thresholding for emotion prediction.
- Careful model and dataset selection.

Applications

- **Chatbots and Virtual Assistants:** Improve emotional responsiveness.
- **Mental Health Monitoring:** Detect signs of distress in user conversations.
- **Customer Feedback Analysis:** Identify customer emotions in reviews.
- **Social Media Monitoring:** Analyze public sentiment during events.
- **Personalized Content Delivery:** Tailor content recommendations based on user emotions.

Future Work

- Fine-tune with larger and more balanced datasets to improve rare emotion detection.
- Expand emotion categories for broader emotional understanding.
- Deploy as a real-time web or mobile application.
- Integrate multilingual support for non-English emotion detection.
- Explore model optimization techniques for faster inference on edge devices.

Conclusion

- Successfully built a lightweight, accurate emotion detection system.
- No retraining necessary.
- Ready for practical applications such as:
 - Chatbots
 - Mental health apps
 - Customer feedback analysis