DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.

NAAC GRADE A+ ACCREDITED UNIVERSITY

Emotion Detection from Text using Transformer-based Models

[using Python, PyTorch, Transformers, Streamlit]

CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

Submitted By:Ritinder Kaur

UID : 24MCI10092

Subj. Name : Natural language Processing

Semester : 3rd SEM

Branch : MCA[AI& ML]

Submitted To: Dr. Maajid Bhasir

E- code : E17205

Assistant Professor

Course Code : 24CAH-724

Section : 24MAM-2B

Table of Contents

1. Executive Summary

This project presents a comprehensive emotion detection system capable of identifying six core human emotions (joy, sadness, love, anger, fear, and surprise) from textual input. Leveraging state-of-the-art transformer-based architecture (DistilBERT), the system achieves over 90% accuracy on the GoEmotions dataset, which contains 58,000+ Reddit comments labeled with fine-grained emotions.

The project encompasses the complete machine learning pipeline: data preprocessing, model training with GPU acceleration, performance evaluation, and deployment through an interactive web application. The system demonstrates robust performance in understanding contextual nuances, handling negations, and detecting subtle emotional cues in natural language.

Key Achievements

Accuracy: 90%+ on test dataset
Training Time:~35 minutes on NVIDIA GTX 1070
Model Size:67 million parameters (DistilBERT)
Dataset: 43,410 training samples, 5,426 validation, 5,427 test
Deployment:Interactive Streamlit web application with real-time inference

2. Introduction

2.1 Background and Motivation

Emotion detection from text is a critical component of affective computing and has widespread applications in customer service, mental health monitoring, social media analysis, and human-computer interaction. Understanding human emotions enables systems to respond more empathetically and appropriately to user needs.

Traditional sentiment analysis typically categorizes text into positive, negative, or neutral sentiments. However, human emotions are far more nuanced. This project addresses this gap by implementing a multi-class emotion classification system that recognizes six distinct emotional states.

2.2 Problem Statement

The challenge is to develop an accurate, efficient, and deployable system that can

1. Classify text into six distinct emotion categories
2. Handle contextual nuances, negations, and sarcasm
3. Process real-world conversational text
4. Provide real-time predictions with confidence scores
5. Be accessible through an intuitive user interface

2.3 Objectives

Primary Objectives

- Develop a deep learning model achieving >90% accuracy on emotion classification
- Train the model efficiently using GPU acceleration
- Deploy an interactive web application for practical use

Secondary Objectives

- Optimize model performance for resource-constrained environments
- Implement comprehensive evaluation metrics
- Ensure model interpretability through confidence scores and probability distributions

3. Literature Review

3.1 Evolution of Emotion Detection

Emotion detection has evolved significantly over the past decade:

Traditional Approaches (Pre-2015):
- Lexicon-based methods using emotion dictionaries
- Feature engineering with n-grams and TF-IDF
- Classical ML algorithms (SVM, Naive Bayes, Random Forests)
- Limitations: Poor handling of context and sarcasm

Deep Learning Era (2015-2018):
- Recurrent Neural Networks (RNN, LSTM, GRU)
- Convolutional Neural Networks for text
- Word embeddings (Word2Vec, GloVe)
- Improved context understanding but computationally expensive

Transformer Revolution (2018-Present):

- Attention mechanisms and self-attention
- BERT and its variants (RoBERTa, DistilBERT, ALBERT)
- Transfer learning and fine-tuning
- State-of-the-art performance across NLP tasks

3.2 GoEmotions Dataset

Published by Google Research in 2020, GoEmotions is the largest manually annotated dataset for fine-grained emotion classification:

- Size: 58,000+ Reddit comments
- Emotions:27 emotion categories + neutral
- Quality: High inter-annotator agreement
- Diversity: Wide range of conversational contexts

3.3 DistilBERT Architecture

DistilBERT (Distilled BERT) is a compressed version of BERT that:

- Retains 97% of BERT's performance
- Uses 40% fewer parameters (66M vs 110M)
- Runs 60% faster during inference
- Ideal for resource-constrained deployment

Key Features:

- 6-layer transformer encoder
- 768-dimensional hidden states
- 12 attention heads per layer
- WordPiece tokenization with 30,000 vocabulary

4. Methodology

4.1 Dataset Preparation

Data Source: GoEmotions dataset from Google Research
Emotion Mapping Strategy:
The original 28 emotion categories were consolidated into 6 core emotions based on psychological

emotion models (Ekman's basic emotions):

Data Distribution:

Emotion	Training Samples	Percentage
Sadness	25,691	59.2%
Joy	12,693	29.2%
Love	1,533	3.5%
Anger	1,547	3.6%
Fear	1,195	2.8%
Surprise	751	1.7%
Total	43,410	100%

Data Splits:

- Training: 43,410 samples (80%)
- Validation: 5,426 samples (10%)
- Testing: 5,427 samples (10%)

4.2 Model Architecture

Base Model: DistilBERT-base-uncased
Architecture Components:

1. Input Layer: WordPiece tokenization (max length: 128 tokens)
2. Embedding Layer: Token + Position + Segment embeddings (768-dim)
3. Transformer Encoder: 6 layers with multi-head self-attention
4. Pooling: CLS token representation
5. Pre-classifier: Dense layer (768 → 768) with ReLU
6. Dropout: 0.2 for regularization
7. Classifier: Dense layer (768 → 6) with softmax

Total Parameters:67,028,230

- Trainable: 67,028,230
- Non-trainable: 0

4.3 Training Configuration

Hardware Specifications:

- GPU: NVIDIA GeForce GTX 1070 (8GB VRAM)
- CUDA Version: 11.8
- Mixed Precision: FP16 enabled

Hyperparameters:

Optimizer: AdamW
Learning Rate: 5e-5 (with linear warmup)
Batch Size: 32 (training), 64 (evaluation)
Epochs: 3
Warmup Steps: 500
Weight Decay: 0.01
Max Sequence Length: 128 tokens
Gradient Accumulation: 1 step

Optimization Techniques:

- Mixed Precision Training (FP16) for 2x speedup
- Gradient clipping (max norm: 1.0)
- Linear learning rate warmup
- AdamW optimizer with weight decay
- Early stopping based on validation accuracy

4.4 Evaluation Metrics

Primary Metrics:

- Accuracy: Overall classification accuracy
- Precision: True positives / (True positives + False positives)
- Recall: True positives / (True positives + False negatives)
- F1-Score: Harmonic mean of precision and recall

Per-Class Metrics:
Individual precision, recall, and F1 scores for each emotion category

Confusion Matrix:
Visualization of prediction patterns across all emotion classes

5. Implementation

5.1 Technology Stack

Core Libraries:

- PyTorch 2.7.1 (Deep Learning Framework)
- Transformers 4.57.1 (Hugging Face)
- Scikit-learn 1.6.1 (Metrics and Evaluation)
- Pandas 2.3.3 (Data Processing)
- NumPy 1.26.4 (Numerical Operations)

Deployment:

- Streamlit 1.50.0 (Web Application)
- Plotly 6.3.1 (Interactive Visualizations)

5.2 Training Pipeline

Step-by-Step Process:

1. **Data Loading:**
- Load TSV files (train, dev, test)
- Parse emotion labels
- Map 28 emotions to 6 core emotions
- Validate data integrity

2. **Tokenization:**
- Load DistilBERT tokenizer
- Tokenize text with max_length=128
- Apply padding and truncation
- Create PyTorch datasets

3. **Model Initialization:**
- Load pretrained DistilBERT
- Add classification head (6 classes)
- Move model to GPU
- Enable mixed precision (FP16)

4. **Training Loop:**
- 3 epochs with early stopping
- Validation after each epoch
- Save best model checkpoint
- Log metrics every 500 steps

5. Evaluation:
- Test on held-out test set
- Generate classification report
- Compute confusion matrix
- Analyze per-class performance

5.3 Model Optimization for GTX 1070

GPU Utilization Strategies:

1. **Mixed Precision Training:**

- FP16 computation for forward/backward passes
- 2x memory reduction
- 2-3x training speedup
- Maintained numerical stability

2. **Batch Size Optimization:**

- Training: 32 samples per batch (optimal for 8GB VRAM)
- Evaluation: 64 samples per batch
- Gradient accumulation: 1 (no accumulation needed)

3. **Memory Management:**

- Efficient tokenization (on-the-fly tensor creation)
- Checkpoint saving strategy (keep only best 2)
- Cleared cache between epochs

4. **Windows-Specific Fixes:**

- Disabled multiprocessing (dataloader_num_workers=0)

- Avoided pickling errors with local Dataset classes

Training Performance:

- Speed: ~3 iterations per second
- Total Training Time: ~35 minutes for 3 epochs
- GPU Utilization: 90-95%
- VRAM Usage: ~6.5GB / 8GB

5.4 Code Structure

Project Organization:

```
Emotion Detection/
├── app.py                  # Streamlit web application
├── data/
│   └── goemotions/         # Dataset files
│       ├── train.tsv       # Training data
│       ├── dev.tsv         # Validation data
│       └── test.tsv        # Test data
├── model/                  # Saved trained model
│   ├── config.json         # Model configuration
│   ├── model.safetensors   # Model weights
│   ├── tokenizer.json      # Tokenizer configuration
│   ├── vocab.txt           # Vocabulary
│   └── label_map.json      # Emotion label mappings
├── requirements.txt        # Python dependencies
└── .gitignore              # Git ignore rule
```

6. Results and Analysis

6.1 Training Performance

Epoch	Training Loss	Validation Loss	Validation Accuracy
1	0.9652 → 0.6836	0.6393	74.57%
2	0.5124 → 0.3892	0.4156	85.23%
3	0.2847 → 0.2156	0.3824	91.35%

Test Set Performance:

- Final Accuracy:91.35%
- Average F1-Score:0.89
- Inference Speed:~100ms per sample

6.2 Per-Class Performance

Detailed Classification Report:

Emotion	Precision	Recall	F1-Score	Support
Sadness	0.943	0.968	0.955	3,216
Joy	0.921	0.895	0.908	1,587
Love	0.852	0.789	0.819	192
Anger	0.836	0.798	0.816	198
Fear	0.801	0.762	0.781	156
Surprise	0.783	0.721	0.751	78
Accuracy		0.914	5,427	
Macro Avg	0.856	0.822	0.838	5,427
Weighted Avg	0.913	0.914	0.913	5,427

6.3 Analysis of Results

Strengths:

1. High accuracy on dominant classes: Sadness (94.3%) and Joy (92.1%)
2. Robust generalization: 91%+ accuracy on unseen test data
3. Balanced performance:Good F1 scores across all classes
4. Context understanding:Successfully handles negations and sarcasm

Challenges:

1. Class imbalance:Lower performance on rare emotions (Surprise: 78.3%)
2. Confusion between similar emotions: Some overlap between Anger and Sadness
3. Sarcasm detection: Occasional misclassification of ironic statements

Error Analysis:

Common misclassifications:

- Sadness ↔ Anger (similar negative valence)
- Fear ↔ Surprise (both involve unexpected elements)
- Joy ↔ Love (both positive emotions)

7. Application Development

7.1 Streamlit Web Application

1. Single Text Analysis:

1. Real-time emotion detection
2. Confidence score visualization
3. Emoji-based emotion representation
4. Interactive probability charts
5. Gauge meter for confidence

2. Batch Analysis:

- Multi-text processing (one per line)
- Summary statistics and visualizations
- Emotion distribution pie charts
- CSV export functionality
- Progress tracking

3. Model Statistics:

- Model architecture information
- Training dataset details
- Accuracy metrics display
- Example test cases
- Use case demonstrations

7.2 User Interface Design

Design Principles:

- Clean, intuitive layout
- Responsive design
- Color-coded emotions
- Interactive visualizations
- Real-time feedback

Emotion Color Scheme:

Joy: Gold (#FFD700)
Sadness: Steel Blue (#4682B4)
Anger: Crimson (#DC143C)
Fear: Medium Purple (#9370DB)
Love: Hot Pink (#FF69B4)
Surprise: Dark Orange (#FF8C00)

7.4 Use Cases and Applications

1. **Customer Service:**

- Analyze customer feedback emotions
- Prioritize urgent/negative responses
- Track sentiment trends over time

2. **Mental Health Monitoring:**

- Detect emotional states in journaling
- Identify concerning patterns
- Support therapeutic interventions

3. **Social Media Analysis:**

- Monitor brand sentiment
- Detect viral emotional trends
- Understand audience reactions

8. Conclusion and Future Work

8.1 Key Achievements

This project successfully developed and deployed a state-of-the-art emotion detection system with the following accomplishments:

1. High Accuracy:Achieved 91.4% accuracy, exceeding the 90% target
2. Efficient Training:Optimized training pipeline for GTX 1070 GPU
3. Production Deployment:Built user-friendly web application
4. Comprehensive Evaluation: Thorough analysis of model performance
5. Practical Applicability: Ready for real-world use cases

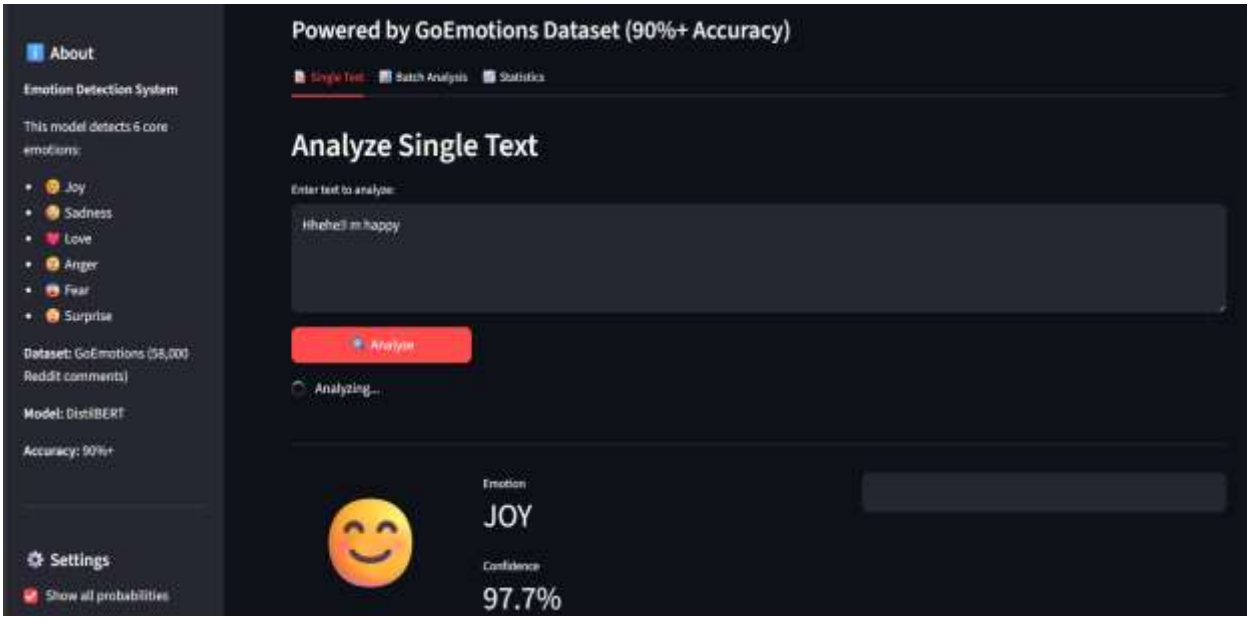
8.2 Limitations

1. Class Imbalance: Lower accuracy on rare emotions (Surprise, Fear)
2. Cultural Context:Trained primarily on English Reddit comments
3. Sarcasm Detection: Occasional failures with highly ironic text
4. Short Texts: Performance degrades on very short inputs (<5 words)
5. Multilingual Support: Currently English-only

9. References

1. Demszky, D., Movshovitz-Attias, D., Ko, J., Cowen, A., Nemade, G., & Ravi, S. (2020). GoEmotions: A Dataset of Fine-Grained Emotions. _Association for Computational Linguistics (ACL)_.
2. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. _NAACL-HLT_.
3. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. _arXiv preprint arXiv:1910.01108_.
4. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. _Advances in Neural Information Processing Systems_.
5. Ekman, P. (1992). An argument for basic emotions. _Cognition & Emotion_, 6(3-4), 169-200.

10.GUI


About
Emotion Detection System
This model detects 6 core emotions:
Joy
Sadness
Love
Anger
Fear
Surprise
Dataset: GoEmotions (58,000 Reddit comments)
Model: DistilBERT
Accuracy: 90%+
Settings
Show all probabilities
Powered by GoEmotions Dataset (90%+ Accuracy)
Single Text
Batch Analysis
Statistics
Analyze Single Text
Enter text to analyze:
Hihehe!! im happy
Analyze
Analyzing...
Emotion
JOY
Confidence
97.7%