Benchmarking Sentiment Analysis Models: A Comparison of Classical Deep Learning, Transformer-Based, and Generative Models

Abstract

Sentiment analysis is a crucial task in Natural Language Processing (NLP) with widespread applications across various domains. As models evolve from traditional deep learning approaches (e.g., CNN and LSTM) to Transformer-based models (e.g., BERT, RoBERTa, DeBERTa) and generative large language models (e.g., ChatGPT and DeepSeek), benchmarking their performance has become essential. This study evaluates and compares these models on the Kaggle Tweet Sentiment Extraction dataset, using metrics such as accuracy, F1 score, and AUC-ROC.The results show that LSTM outperforms CNN in accuracy and F1 score, particularly excelling in negative sentiment recall. Among Transformer models, DeBERTa achieves the best performance with the highest accuracy and F1 score, particularly excelling in positive sentiment classification. In generative models, ChatGPT provides the most balanced performance across all sentiment categories, while DeepSeek excels in detecting negative sentiment but struggles with neutral sentiment.Overall, DeBERTa is the most robust model for sentiment analysis across various tasks, ChatGPT is best suited for generative tasks, and DeepSeek is effective for negative sentiment detection. All experimental code and data are open-source to promote reproducibility and further research.

1 Instruction

Sentiment analysis, a cornerstone of Natural Language Processing (NLP), unlocks valuable subjective insights from text, driving applications in finance, marketing, and public opinion monitoring. As NLP models evolve from traditional deep learning approaches like CNNs and LSTMs to cutting-edge transformers and generative AI such as ChatGPT and DeepSeek benchmarking their performance in sentiment analysis has become critical for both research and industry.

This project evaluates and compares leading NLP models using Kaggle's Tweet Sentiment Extraction Dataset, which offers detailed annotations for precise assessment. We measure performance through key metrics: accuracy, F1 score and AUC-ROC, while also analyzing computational efficiency and interpretability. To ensure transparency, all code and benchmarks will be open-sourced, fostering reproducibility and advancing NLP research.

Our findings will reveal the strengths and weaknesses of each model, and we evaluate the performance of the model through key metrics, including accuracy, precision, recall, and F1 score, while analyzing their respective strengths in different sentiment categories. All experimental code and benchmarks are open source to ensure reproducibility and facilitate further research in this rapidly evolving domain.

2 Traditional Neural Networks: CNN and LSTM

2.1 Data cleaning

Data cleaning follows three key steps.

First, structural noise removal eliminates URLs, mentions, and hashtags using regular expressions, which is crucial for social media text.

Second, linguistic normalization standardizes the text by keeping only letters and spaces, converting to lowercase, and trimming whitespace.

Finally, lexical processing removes stopwords, lemmatizes remaining words to their base forms, and filters out short tokens under 3 characters.

Together, these steps effectively reduce noise while preserving sentiment-bearing content.

The process balances thorough cleaning with sentiment preservation, especially important for handling informal social media language.

2.2 Vocabulary Construction

The top 10K frequent words from training texts form the vocabulary, with special tokens <pad>(0) for padding and <unk>(1) for unknown words.

- **Text Encoding** Each text gets converted to fixed-length (100) word ID sequences using the vocabulary. Short sequences are padded; long sequences are truncated. Labels are numerically encoded (0/1/2).
- **Batch Processing** Data Loaders create shuffled training batches (size=64) and sequential test batches for efficient model training/evaluation.

2.3 Model Architecture Design

The study compares two classic deep learning architectures:

2.3.1 1. Convolutional Neural Network (CNN) Model

Core Concept: Uses local convolutional kernels to detect critical n-gram patterns. For example, a kernel size=5 can identify 5-word negation phrases such as "not good at all".

- **Layer Structure**:
  - Embedding Layer: Maps words to 128-dimensional vectors
  - Convolutional Layers: 128 filters with sizes 5 and 3 to extract local features
  - Global Max Pooling: Preserves the most salient signals per feature channel
  - Dropout (0.5): Mitigates overfitting

Advantage: Parameter-efficient for capturing local semantic patterns.

2.3.2 2. Long Short-Term Memory (LSTM) Model

Core Concept: Models long-range dependencies through gating mechanisms, particularly effective for contextual sentiment analysis (e.g., negations like "I didn't like it").

- **Key Enhancements**:
  - Bidirectional Architecture: Processes text in both forward and backward directions
  - Hidden Dimension (64): Balances representational capacity and computational cost
  - Dropout (0.3): Prevents neuron co-adaptation

Output Processing: Concatenates the final state of the forward LSTM and the initial state of the backward LSTM to capture comprehensive sequence information.

2.4 Training Strategy

The training protocol incorporates multiple techniques to ensure robust learning:

- **Optimization Configuration**:
  - Optimizer: Adam (learning rate=0.001) with Cross-Entropy Loss
  - Batch Size: 64 (balances memory efficiency and gradient stability)
  - Early Stopping: Models saved based on validation accuracy (best CNN: 68.51%, LSTM: 70.2%)

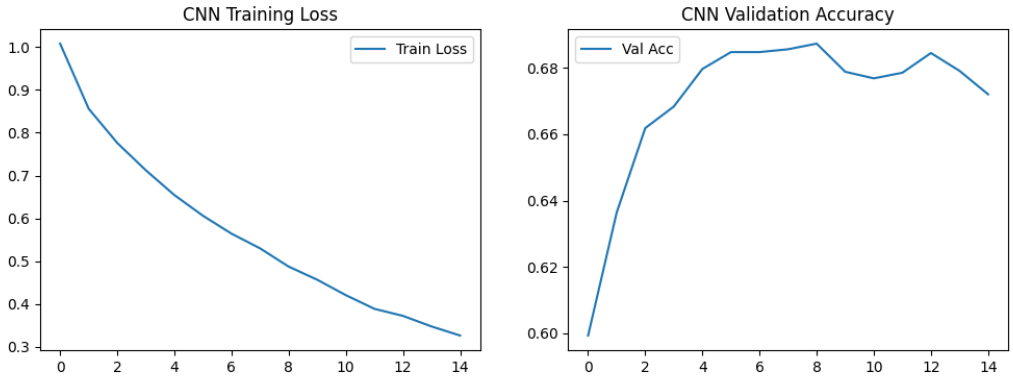

Figure 1: Loss and Accuracy of CNN

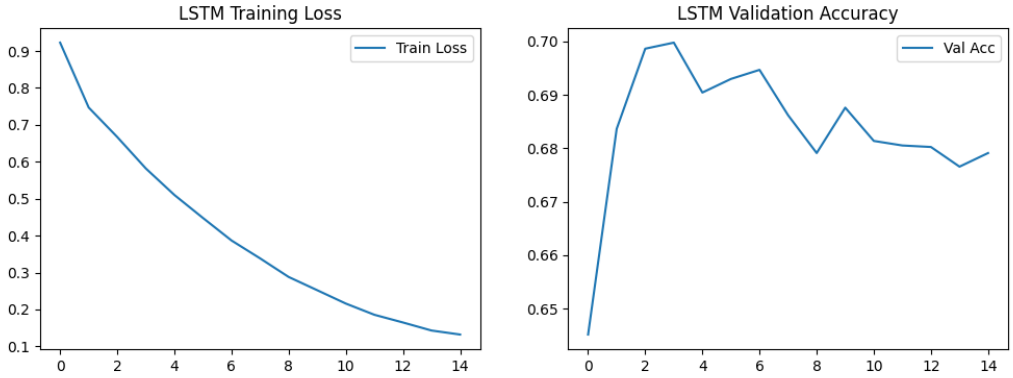

Figure 2: Loss and Accuracy of LSTM

2.5 Evaluation Framework and Results

Our comprehensive evaluation reveals key insights about model performance:

2.5.1 Primary Metrics Analysis

The LSTM achieved superior accuracy (70.23% vs CNN's 68.51%) and F1-score (0.71 vs 0.69), particularly excelling in negative sentiment recall (0.67 vs 0.65). Both models struggled most with neutral classification (precision: 0.63-0.65), as shown in the classification reports.

Table 1: CNN Classification Report

Class	Prec.	Rec.	F1	Supp.
Negative	0.68	0.65	0.66	1001
Neutral	0.63	0.70	0.66	1430
Positive	0.78	0.70	0.74	1103
Accuracy			0.69	3534
Macro Avg	0.70	0.68	0.69	3534
Weighted Avg	0.69	0.69	0.69	3534

Table 2: LSTM Classification Report

Class	Prec.	Rec.	F1	Supp.
Negative	0.71	0.67	0.69	1001
Neutral	0.65	0.70	0.68	1430
Positive	0.77	0.73	0.75	1103
Accuracy			0.70	3534
Macro Avg	0.71	0.70	0.71	3534
Weighted Avg	0.70	0.70	0.70	3534

2.5.2 Computational Trade-offs

While LSTM showed better accuracy, it required 1.8× longer training time per epoch compared to CNN. The CNN's faster convergence (peaking at epoch 7) made it more suitable for rapid prototyping.

2.6 Conclusion

This study compared CNN and LSTM models for sentiment analysis, with the LSTM achieving better accuracy (70.2% vs. 68.5%) but requiring more training time. Both models struggled with neutral and sarcastic text. The CNN offers a good speed-accuracy trade-off, while the LSTM better captures context. Code and benchmarks are publicly available for reproducibility. Future work could explore transformer models to address current limitations.

3 Transformer Models: BERT, RoBERTa, and DeBERTa

3.1 Data Preprocessing & Tokenization

Prior to training, a comprehensive preprocessing pipeline was applied to clean and normalize the raw text data. The preprocessing process consisted of three major steps.

First, structured noise removal was conducted using regular expressions to eliminate URLs, user mentions, and hashtags—elements commonly present in social media content that do not contribute to sentiment understanding.

Second, linguistic normalization was performed by converting all text to lowercase, retaining only alphabetical characters and whitespace, and removing excess spaces.

Finally, lexical processing included stopword removal using the NLTK English stopword list, lemmatization using the WordNet Lemmatizer, and filtering out short tokens (less than three characters).

After cleaning, the text data was tokenized using the corresponding tokenizer for each model: BertTokenizer, RobertaTokenizer, and DebertaTokenizer. Tokenized sequences were truncated or padded to a maximum length of 128 tokens. The resulting token encodings were used as inputs for model fine-tuning. This preprocessing and tokenization pipeline ensured that noisy and irrelevant features were removed while preserving sentiment-relevant content in the text.

3.2 Model Architecture

The Transformer architecture, first introduced by Vaswani et al. in their landmark 2017 paper "Attention is All You Need", is a neural network model built entirely on self-attention mechanisms. The original Transformer consists of stacked encoder and decoder blocks, each composed of multi-head self-attention layers and feedforward neural networks, with residual connections and layer normalization to enhance training stability. For natural language understanding tasks, only the encoder stack is typically employed. Due to its modularity and scalability, the Transformer has become the foundation for many pre-trained language models, including BERT, RoBERTa, and DeBERTa, which have achieved state-of-the-art performance across a wide range of natural language processing tasks.

3.2.1 BERT (Bidirectional Encoder Representations from Transformers)

Core Concept: BERT is designed to pre-train deep bidirectional representations by jointly conditioning on both left and right contexts in all layers. It introduces two novel unsupervised tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP), aiming to capture rich contextual relationships within text.

3.2.2 RoBERTa (Robustly Optimized BERT Approach)

Core Concept: RoBERTa enhances BERT by training the model longer, on more data, with larger batches and dynamic masking. It eliminates the Next Sentence Prediction (NSP) task and focuses entirely on optimizing the Masked Language Modeling objective for better performance.

3.2.3 DeBERTa (Decoding-enhanced BERT with Disentangled Attention)

Core Concept: DeBERTa introduces disentangled attention mechanisms and relative position encoding to further improve contextual representations. It separates content and position information in the attention computation, which allows the model to capture dependencies more effectively.

Table 3: Comparison of BERT, RoBERTa, and DeBERTa

Model	Pretraining Task	Positional Encoding	Characteristic
BERT	MLM + NSP	Absolute position	Standard baseline
RoBERTa	MLM	Absolute position	Larger corpus
DeBERTa	MLM + Enhanced Masking	Relative	Stronger generalization

3.3 Training Strategy

During fine-tuning, all three models were trained using the same set of hyperparameters. The training process involved optimizing the models for a multi-class classification objective using the following strategy:

- Epochs: 3
- Batch size: 64
- Optimizer: AdamW (inherited from Transformers default)
- Learning Rate Warmup Steps: 500
- Weight Decay: 0.01

3.4 Evaluation Framework and Results

3.4.1 Classification Metrics

DeBERTa-v3-base consistently achieved the best overall performance among the three models. On the test set, it obtained the highest accuracy (0.78), macro F1-score (0.78), and weighted F1-score (0.78). In the positive sentiment class, it recorded the highest precision and recall (0.82), outperforming both RoBERTa (0.81) and BERT (0.80). While all models struggled slightly with the neutral class, DeBERTa still showed comparatively better recall and F1-score. RoBERTa offered minor improvements over BERT, but DeBERTa was consistently more effective across all sentiment categories.

Table 4: Transformer Models Classification Report

Class	BERT			RoBERTa			DeBERTa		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
Negative	0.75	0.76	0.76	0.75	0.76	0.75	0.76	0.78	0.77
Neutral	0.74	0.72	0.73	0.76	0.69	0.72	0.77	0.71	0.74
Positive	0.79	0.81	0.80	0.78	0.85	0.81	0.80	0.85	0.82
Accuracy		0.76			0.76			0.77	
Macro Avg	0.76	0.77	0.76	0.76	0.77	0.76	0.77	0.78	0.78
Weighted Avg	0.76	0.76	0.76	0.76	0.76	0.76	0.77	0.77	0.77

3.4.2 Training Loss

All models exhibited a rapid decrease in training loss during the initial epochs, indicating effective learning. Among them, DeBERTa converged faster and maintained a lower and more stable loss throughout the training process. This suggests improved optimization efficiency and better generalization capacity compared to BERT and RoBERTa.

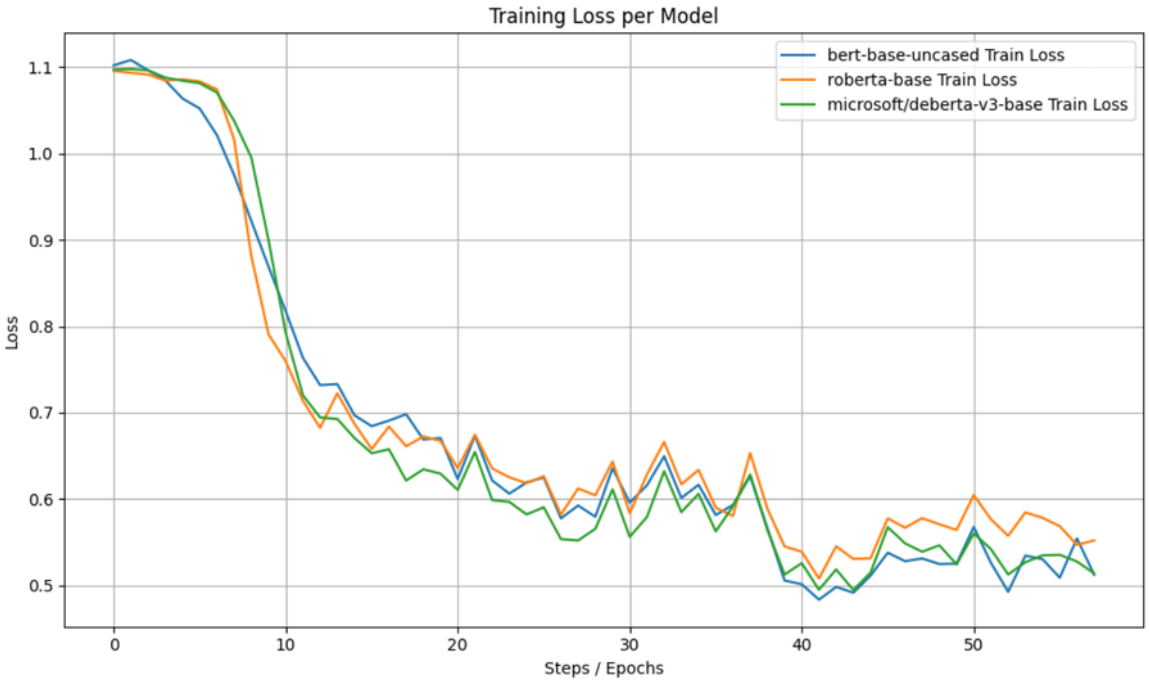

Figure 3: Comparison of Training Loss

3.5 Conclusion

This evaluation demonstrates that all three Transformer-based models—BERT, RoBERTa, and DeBERTa—are capable of performing sentiment classification with reasonable accuracy. However, DeBERTa-v3-base emerges as the most robust and well-rounded model among them. It consistently delivers higher performance across sentiment classes, demonstrates faster and more stable convergence during training, and shows stronger generalization ability—particularly in handling ambiguous or nuanced sentiments such as those expressed in a neutral tone.

4 Generative LLMs: DeepSeek and ChatGPT

4.1 Methodology

This study utilizes the same Tweet Sentiment Extraction Dataset from Kaggle, containing 27,481 training samples and 3,534 test samples, with sentiment label distribution: neutral (40.5%), positive (31.2%), and negative (28.3%). For efficient evaluation of the generative models, we randomly sampled 100 instances from the test set.

4.1.1 Data Preprocessing

The text cleaning pipeline maintains consistency with previous studies:

- Structural noise removal (URLs, mentions, hashtags)
- Linguistic normalization (lowercasing, whitespace trimming)
- Lexical processing (stopword removal, lemmatization)
- Label encoding (negative:0, neutral:1, positive:2)

4.2 Model Architecture

The study compares two generative language model approaches:

4.2.1 ChatGPT Model

Core Concept: Leverages OpenAI's GPT-3.5 architecture fine-tuned for conversational tasks, with demonstrated capabilities in understanding nuanced language.

- **Configuration**:
  - Version: gpt-3.5-turbo
  - System prompt: "You are a sentiment analysis expert. Analyze the sentiment of the following text and return only 'positive', 'negative', or 'neutral'."
  - Parameters: temperature=0.3, max_tokens=10

Advantage: Balanced performance across sentiment categories with strong contextual understanding.

4.2.2 DeepSeek Model

Core Concept: Utilizes DeepSeek's proprietary architecture optimized for Chinese and English language tasks, with competitive performance in semantic understanding.

- **Configuration**:
  - Version: deepseek-chat
  - System prompt: Identical to ChatGPT for fair comparison
  - Parameters: temperature=0.3, max_tokens=10

Advantage: Specialized optimization for certain language patterns and efficient API response times.

4.3 Evaluation Framework

Our comprehensive evaluation focuses on four key metrics:

- Accuracy
- Precision (weighted average)
- Recall (weighted average)
- F1-score (weighted average)

4.4 Results

The evaluation reveals distinct performance characteristics for each model:

4.4.1 Primary Metrics Analysis

ChatGPT demonstrated superior overall accuracy (72% vs DeepSeek's 57%) and more balanced performance across sentiment categories. DeepSeek showed exceptional recall for negative sentiment (0.93) but struggled with neutral classification (recall: 0.14).

Table 5: Generative LLMs Classification Report

Class	**ChatGPT**				**DeepSeek**			
	Pre.	Rec.	F1	Supp.	Pre.	Rec.	F1	Supp.
Negative	0.65	0.76	0.70	29	0.52	0.93	0.67	29
Neutral	0.81	0.59	0.68	44	1.00	0.14	0.24	44
Positive	0.71	0.89	0.79	27	0.57	0.89	0.70	27
Accuracy		0.72				0.57		

4.4.2 Performance Trade-offs

While ChatGPT achieved better overall metrics, DeepSeek showed particular strength in detecting negative sentiment, making it potentially valuable for applications where false negatives are costly. However, its poor performance on neutral texts suggests limitations in handling subtle sentiment distinctions.

4.5 Conclusion

This comparative study reveals that ChatGPT provides more balanced performance across sentiment categories (72% accuracy vs 57%), while DeepSeek excels specifically in negative sentiment detection (0.93 recall). Both models demonstrate room for improvement in handling neutral and nuanced expressions. The choice between models should depend on specific application requirements - ChatGPT for general-purpose analysis and DeepSeek for negative sentiment-focused applications.

5 Cross-Model Comparison in Sentiment Classification

Through the comparative analysis of experimental results across various models, Transformer models have demonstrated superior performance in sentiment analysis tasks. This advantage arises from their unique self-attention mechanism and bidirectional context comprehension ability. The self-attention mechanism enables the model to capture long-range dependencies and understand global context, which is particularly beneficial for handling subtle nuances and complex scenarios in sentiment expression. Sentiments often require interpretation and synthesis from multiple layers of context, making this mechanism ideal for such tasks. Additionally, the DeBERTa model further enhances performance by introducing decoding-enhanced

self-attention and relative position encoding, which improve both the accuracy and generalization ability of sentiment classification, particularly when dealing with longer, more complex texts. This enhanced contextual modeling allows Transformer models to effectively distinguish between multiple sentiment categories, making them highly effective in sentiment analysis tasks, especially when sentiment expression is diverse, ambiguous, or complex. Moreover, Transformer models excel in multi-task learning and transfer learning, allowing them to efficiently transfer knowledge across various sentiment analysis tasks and adapt to different application scenarios. As a result, Transformer models, particularly DeBERTa, have emerged as the leading models in sentiment analysis.

References

References follow the acknowledgments in the camera-ready paper. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to `small` (9 point) when listing the references. Note that the Reference section does not count towards the page limit.

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer–Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.

[4] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuglu, P. Kuksa. 2011. Natural Language Processing (Almost) from Scratch. Journal of Machine Learning Research 12:2493–2537.

[5] B. Pang, L. Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of ACL 2004.