GENERATIVE AI & DEEP LEARNING PROJECT REPORT

Reel Insights: Unveiling Movie Sentiment with Generative AI

SUBMITTED BY

DEBARGHYA CHOWDHURY – 12022002018006

SOUMYAJIT SAPUI – 12022002018067

SAMRAT MONDAL – 12022002018071

SARASIJ DAS – 12022002018072

SUPERVISED BY

Professor Swarnendu Ghosh

INSTITUTE OF ENGINEERING & MANAGEMENT
(UNDER MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY)

APRIL, 2025

Project Tasks:

1. **Problem Definition and Objective Setting**
   - Identify the need for analyzing movie sentiment.
   - Define project goals: accurate sentiment extraction using generative AI techniques.
2. **Literature Review**
   - Study existing sentiment analysis approaches (lexicon-based, machine learning, deep learning).
   - Review generative AI models used in text analysis (e.g., GPT, BERT, LLaMA, etc.).
3. **Dataset Collection and Preprocessing**
   - Collect movie reviews (IMDb, Rotten Tomatoes, etc.).
   - Clean and preprocess data (remove stop words, punctuation, perform tokenization, lemmatization).
4. **Exploratory Data Analysis (EDA)**
   - Analyze sentiment distribution.
   - Identify frequent keywords, phrases, or n-grams.
   - Visualize data trends (word clouds, sentiment histograms).
5. **Model Selection and Development**
   - Select appropriate generative AI models (e.g., GPT, T5, or LLaMA for sentiment generation or classification).
   - Fine-tune models on labeled movie review datasets.
   - Implement training pipeline with relevant frameworks (Hugging Face Transformers, PyTorch, etc.).
6. **Sentiment Interpretation and Generation**
   - Generate textual sentiment summaries or review synthesis.
   - Highlight pros and cons based on generative model output.
   - Optionally, classify sentiments as positive, negative, or neutral..
7. **Conclusion and Future Scope**
   - Summarize key findings.
   - Suggest improvements like multilingual support, sarcasm detection, or video sentiment from trailers

Dataset Name:

IMDb Movie Review Dataset

Source: https://www.tensorflow.org/datasets/catalog/imdb_reviews

Dataset Size: 129.83MiB

Split	Examples
Test	25000
Train	25000
Unsupervised	50000

The dataset is provided in **CSV format** with the following columns:

Column Name	Description
review	The full text of the movie review
sentiment	The sentiment label (positive/negative)

Preprocessing Steps:

To prepare the dataset for model training and evaluation, the following preprocessing steps are applied:

- Removal of HTML tags and special characters
- Lowercasing of all text
- Tokenization of sentences and words
- Removal of stop words
- Lemmatization for reducing words to their base form
- Optionally, conversion into embedding or token IDs for input into generative models.

Advantages of this Dataset:

- Clean and well-structured
- Balanced class distribution
- High-quality human-written reviews
- Suitable for both classification and generative tasks

Model Name:

GPT-2 (or GPT-Neo / T5 based on implementation)

Model Architecture:

Component	Details
Type	Transformer-based Generative Language Model
Layers	12 (for GPT-2 Small)
Hidden Size	768
Attention Heads	12
Parameters	~124M (GPT-2 Small)
Tokenizer	Byte Pair Encoding (BPE)
Input Format	Tokenized text (prompt + review)
Output	Generated text and/or sentiment classification

Training Configuration:

Parameter	Value
Epochs	3–5
Batch Size	8–32 (depending on GPU)
Learning Rate	5e-5
Max Sequence Length	512 tokens
Hardware	GPU-enabled environment

Training Process:

Dataset Split:

- Training set: 80%
- Validation set: 10%
- Test set: 10%

Model Configuration

- **Base Model:** GPT-2 (or T5-small)
- **Framework:** PyTorch/TensorFlow with Hugging Face Transformers
- **Training Mode:** Causal Language Modeling (for GPT) or Sequence-to-Sequence (for T5)

Training Loop

1. **Tokenize Inputs:** Encode the input review and target label.
2. **Forward Pass:** Pass input through the model.
3. **Loss Calculation:** Compute loss between predicted tokens and ground truth.
4. **Backpropagation:** Compute gradients and update model weights.
5. **Validation:** Evaluate performance on validation set after each epoch.
6. **Checkpoint:** Save model checkpoints at regular intervals.

Evaluation after training

- Evaluate model performance on unseen test data.
- Generate classification labels or sentiment explanations.
- Compare results with baseline models (e.g., SVM, LSTM).

Results and Analysis: