LLMs with an Opinion

E6691.2025Spring.IDLI.report.si2468.app4617.rpp2142
Sriraman Iyengar si2468, Anushka Pachaury ap4617, Radhika Patel rpp2142
Columbia University

Abstract

This project investigates sentiment-controlled movie review generation using large language models (LLMs), specifically GPT and LLaMA-2, fine-tuned on an enriched version of the IMDb dataset. The primary goal was to condition review generation on user-specified sentiment and movie metadata using instruction-tuned training. To support this, we curated a novel dataset by extracting movie names and generating summaries from raw reviews using the OpenAI API. We fine-tuned both models using a masked loss strategy that computed gradients only on the review portion of instruction-summary-review triplets. Multiple prompting strategies were evaluated, and inference was conducted using both pretrained and fine-tuned models under two prompt types: one with sentiment and movie name, and one augmented with a movie summary. Controlled experiments were also conducted to study the effect of decoding parameters such as temperature, top-k, and top-p on output quality. The key challenge was aligning generated content with both sentiment intent and reference semantics. Our approach significantly improved sentiment match accuracy (from 53% to 90% in LLaMA-2 and from 54% to 98% in GPT) and semantic similarity scores. The original objective to generate high-quality, sentiment-aligned reviews conditioned on structured prompts was successfully met.

1. Introduction

With the rapid advancement of large language models, the ability to generate human-like text has opened up new possibilities across domains such as content creation, summarization, and recommendation systems. One exciting application is the automated generation of movie reviews, which can help users make informed viewing choices while reducing manual effort in content curation. However, ensuring that generated reviews are not only coherent but also semantically accurate and sentimentally appropriate presents a non-trivial challenge.

This project aims to develop a two-stage NLP pipeline that automatically generates movie reviews and evaluates their sentiment and semantic correctness. We fine-tune GPT-2, a powerful generative language model, to produce textual reviews conditioned on movie-related prompts [3]. Additionally, we incorporate LLaMA-2 7B, a more recent and larger generative model from Meta, to serve as a comparative model. This allows us to assess differences in output quality, coherence, and alignment between the two models. To assess the quality of the generated text, we use BERT, a language model trained for language understanding, to perform sentiment classification and semantic evaluation [4].

A major technical challenge in this task is ensuring that the output from generative models not only flows naturally but also aligns with expected sentiment (e.g., a positive review for a well-rated movie). Generative models, while fluent, are prone to hallucinations or off-topic responses. By pairing GPT-2 and LLaMA-2 with BERT in a feedback loop, we seek to mitigate these issues and improve both content quality and relevance.

Our approach combines generation and evaluation to produce movie reviews with validated sentiment. This project demonstrates the potential for integrating large pretrained models in real-world text generation applications and highlights the strengths of decoder and encoder-based transformers.

2. Background

In recent years, large-scale language models have grown capable of understanding and generating human-like text. This progress has been influenced by access to massive datasets, increases in computational power, and the introduction of the transformer architecture, which replaced recurrence with self-attention mechanisms [5]. Transformers enabled more effective modeling of long-range dependencies in text and significantly improved performance across a wide range of NLP tasks, including translation, summarization, and question answering.

Building on this foundation, a new class of language models emerged - ones that are pretrained on massive corpora and later fine-tuned for specific downstream tasks. These models use self-supervised learning objectives, such as predicting masked words or the next word in a sequence, to learn general-purpose language representations. By decoupling from task-specific learning, this approach allows models to generalize well and transfer knowledge across tasks, even with limited labeled data.

At the same time, the field has seen a growing interest in combining generative capabilities with mechanisms for evaluation and control. While generative models can produce fluent and diverse text, they may occasionally generate content that is off-topic, semantically inconsistent, or sentimentally misaligned. To mitigate this, researchers have explored the use of discriminative models, such as those designed for sentiment classification or semantic similarity, to guide, evaluate, or filter the output of generative systems. This hybrid strategy ensures that the generated text is not only linguistically fluent but also contextually appropriate and emotionally consistent.

This two-part approach (generation followed by evaluation) forms the foundation of our project. In the following sections, we explore how GPT-2, a decoder-only language model, is used to generate movie reviews, and how BERT, an encoder-only model, is employed to evaluate the semantic alignment and sentiment of those reviews. Together, these models enable a robust system for automatic review generation and sentiment assessment.

GPT-2

The original GPT-2 paper, *Language Models are Unsupervised Multitask Learners*, demonstrated that large-scale language models trained with a simple next-word prediction objective can achieve strong performance across a variety of tasks without task-specific supervision [3]. The authors trained GPT-2 on a newly collected dataset called WebText, comprising over 8 million web pages. The model, which consists of 1.5 billion parameters, learns to generate coherent, contextually appropriate text by leveraging the Transformer decoder architecture and autoregressive training.

A key insight from the paper is that scaling up both the model size and training data leads to increasingly strong zero-shot performance. Without using any labeled data or fine-tuning, GPT-2 was able to perform competitively on tasks such as translation, question answering, and summarization, simply by conditioning on prompts that frame the problem. For example, adding a phrase like "Translate English to French:" allowed the model to generate reasonable translations, even though it had never seen labeled translation pairs during training.

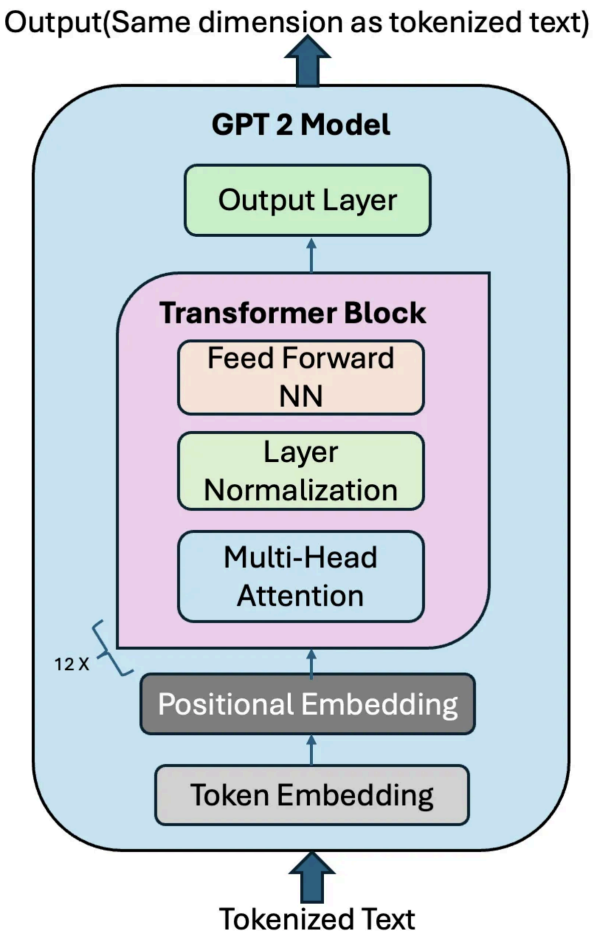

Figure 1: High-level overview of GPT-2's decoder-only architecture.

LLaMA-2 7B

LLaMA-2 is a family of transformer-based language models introduced by Meta. The 7B variant refers to the model with approximately 7 billion parameters, making it significantly larger and (presumably) more capable than GPT-2, while remaining the most lightweight in the LLaMA-2 series. LLaMA-2 models are trained on a diverse mixture of publicly available data sources and are designed to be efficient, performant, and accessible for academic and research purposes.

Like GPT-2, LLaMA-2 7B follows a decoder-only transformer architecture, which makes it suitable for generative tasks such as text completion and response generation. However, LLaMA-2 benefits from advancements in training stability, tokenizer improvements, and dataset diversity. These enhancements contribute to its ability to generate more coherent and contextually appropriate output, even without extensive fine-tuning. In practice, LLaMA-2 has been shown to

achieve performance on par with or exceeding other closed models of comparable size.

In this project, we fine-tune LLaMA-2 7B on movie-related prompts to serve as an alternative to GPT-2 for review generation. This provides a meaningful comparison between an earlier generation model (GPT-2) and a more modern, larger-scale generative model. By comparing the outputs of both systems and evaluating sentiment and semantic analysis, we aim to understand the tradeoffs between model size, fluency, alignment, and ease of fine-tuning.

BERT

BERT (Bidirectional Encoder Representations from Transformers), introduced by Devlin et al. in 2018, marked a significant breakthrough in natural language processing by pretraining deep bidirectional representations from unlabeled text. Unlike earlier models that processed text left-to-right or right-to-left, BERT uses a Transformer encoder to capture context from both directions simultaneously. This bidirectional architecture allows BERT to understand the full context of a word based on all of its surroundings, rather than just the preceding words. The model was pretrained on a large amount of data, including English Wikipedia and BookCorpus, using two self-supervised tasks: masked language modeling (MLM) and next sentence prediction (NSP).

The masked language modeling task involves randomly masking out tokens in a sentence and training the model to predict them, which encourages BERT to develop a deep understanding of syntax and semantics. The next sentence prediction task helps the model understand relationships between sentences, which is crucial for downstream tasks such as question answering and natural language inference. After this pre-training, BERT can be fine-tuned on specific tasks by adding a small output layer and training on labeled datasets. This two-stage training process, pre-training followed by fine-tuning, has become the foundation of many state-of-the-art NLP systems, and allows BERT to be extended to relevant tasks like sentiment analysis and semantic similarity.

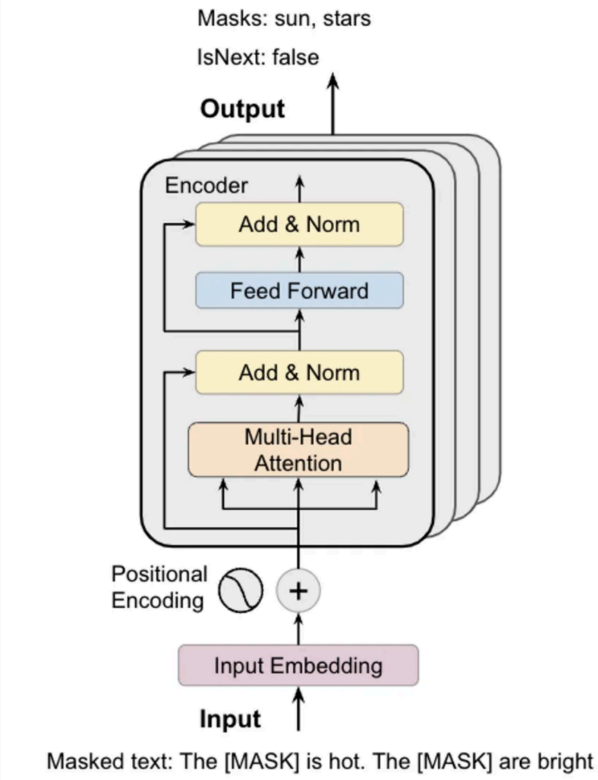

Figure 2: BERT's encoder-only architecture

Since BERT is an encoder-only model, it is not suitable for text generation. In contrast, the decoder-only architecture of GPT-2 is more effective for generative tasks. Thus, we hope to include these models in our tasks as follows:

1) Fine-tune GPT to generate movie reviews.
2) Fine-tune LLaMA-2 to generate movie reviews.
3) Evaluate the reviews (semantically) using BERT.

These two models can be used in conjunction to create a system in which movies can be reviewed, and those reviews can be assigned a sentiment (positive vs negative).

Comparison to Prior Work

Our methodology draws partial inspiration from Stilwell (2024), *"Explainable Prompt Learning for Movie Review Sentiment Analysis"*, which fine-tunes LLaMA and GPT-style models using prompt learning for classification, incorporating explainability techniques such as SHAP and LIME to interpret model decisions [10]. In contrast, our project is focused on generation, not classification, and we emphasize autoregressive instruction learning with masked loss to train models to

conditionally generate movie reviews based on structured prompts.

Stilwell evaluates classification models primarily using sufficiency and faithfulness scores to assess interpretability. Our work, by comparison, evaluates generated text using generative quality metrics (BLEU, ROUGE-L), semantic similarity, and sentiment accuracy under varied decoding settings. Additionally, we do not attempt to explain model decisions but instead focus on improving conditional generation through instructional prompt design and decoding strategy tuning. As our study does not seek to reproduce, extend, or benchmark against classification accuracy results, we do not compare our outcomes to those reported by Stilwell, since the underlying task formulations are fundamentally different.

3. Methodology of our Project

In this project, we investigated sentiment-conditioned review generation by fine-tuning and evaluating LLMs, specifically GPT and LLaMA-2, on a curated and augmented version of the IMDb movie review dataset. Our approach integrates instruction tuning, prompt variation, and decoding strategy exploration to study how these factors influence the quality, sentiment alignment, and semantic similarity of generated movie reviews.

Dataset Curation and Augmentation

We began with the publicly available IMDb movie review dataset from Hugging Face, consisting of labeled training and test samples. For each review in the dataset, we extracted the corresponding movie title using pattern-based parsing. We then used the OpenAI GPT-4 API to generate a summary for each movie based on its title and associated review. These summaries were used to construct instruction-based inputs for fine-tuning GPT-2.

While the original IMDb dataset is sufficient for training sentiment classifiers or decoder-only models in a generic setting, it was not directly suitable for instruction tuning GPT-2. Unlike more recent instruction-tuned models such as LLaMA-2, GPT-2 was pre-trained without exposure to instruction-following data, making it poorly equipped to understand vague or open-ended prompts like “Criticize this movie.” As a result, simply prepending sentiment instructions to raw reviews—as done in our LLaMA-2 fine-tuning pipeline—was ineffective for GPT-2. To enable meaningful instruction tuning, we needed to explicitly ground each prompt with contextual cues, such as the movie title and a summary. This additional information allowed us to craft semantically rich and directive prompts (e.g., “Explain why *Inception* was excellent. Summary: …”), helping the model associate sentiment with content.

GPT Instruction-Tuning

We applied full-parameter instruction fine-tuning to GPT using instruction-summary-review triplets, on a dataset constructed with the following structure:

Prompt (Template with Movie Name + Sentiment) + Summary: {Movie Summary}
→ Target: {Original IMDb Review}

Input prompts were generated from a set of handcrafted sentiment-conditioned templates. Examples include:

***Positive Prompts**:*
“*Write a positive movie review for the movie '{movie}' given the following summary:"*
"Explain why '{movie}' was excellent. Summary:"

***Negative Prompts**:*
"Criticize the movie '{movie}'. Summary:"
"Describe why '{movie}' is disappointing. Summary:"

The input prompt included the sentiment label, movie title, and optionally the summary. These prompts were randomly sampled to ensure diversity. Our goal was to train GPT to generate sentiment-aligned reviews conditioned on natural language instructions.

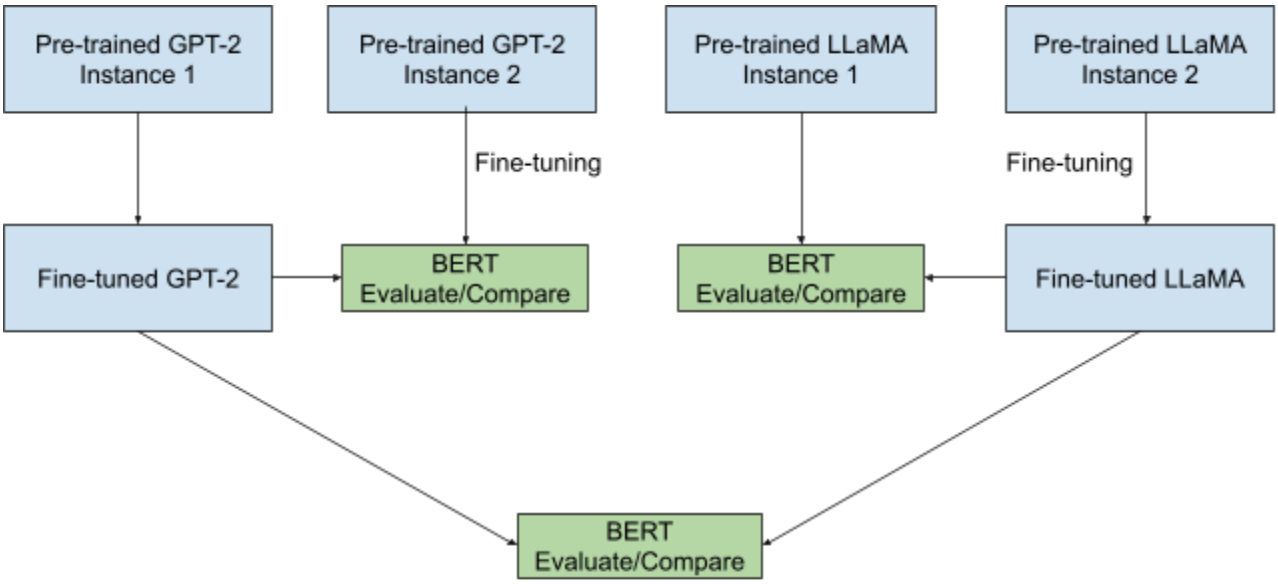

Figure 3: Basic flowchart for fine tuning process. For both GPT-2 and LLaMA, pretrained instances will be compared with fine-tuned instances to assess impact of the fine-tuning process. Fine-tuned versions of both models will be compared with each other to assess impact of model type for this task. This comparison will be of the evaluations of each individual fine-tuned instance on the testing set for a variety of metrics, such as sentiment and BLEU scores.

Each prompt included the movie name and was concatenated with the generated summary to serve as the model input, while the original IMDb review served as the output. Training followed a causal language modeling (CLM) objective with instruction masking—token labels corresponding to the prompt portion were set to -100 to prevent gradient updates, allowing the model to focus on learning to generate the review. Fine-tuning was performed using the Hugging Face Trainer API for 3 epochs with a per-device batch size of 2, constrained by available GPU memory. Inputs were tokenized to a maximum length of 512 tokens with padding and truncation enabled. We used mixed precision training where possible and set evaluation checkpoints every 500 steps, with logging every 100 steps.

LLaMA-2 Fine-Tuning

We fine-tuned the LLaMA-2 7B model using a resource-efficient training strategy known as QLoRA (Quantized Low-Rank Adapter), which enables fine-tuning of large models on limited GPU memory. The base model (meta-llama/Llama-2-7b-hf) was quantized to 4-bit precision, and LoRA adapters were added to enable low-rank parameter updates.

The training dataset was an augmented version of the IMDb movie review corpus. Instead of using explicit movie names or summaries, we constructed instruction-response pairs in which the instruction was a natural language prompt requesting either a positive or negative review. These prompts were designed to simulate diverse, open-ended instructions and were randomly selected from handcrafted templates such as:

Positive Prompts:
"Write a positive movie review."
"What made this movie so enjoyable?"

Negative Prompts:
"Criticize this movie:"
"Explain why the movie was terrible:"

This approach was inspired by a prior GPT-2 experiment, in which we used similarly vague prompts without summaries or movie names. That earlier setup performed poorly, likely due to GPT-2's limited capacity for abstraction. In contrast, we hypothesized that LLaMA-2's larger model capacity and instruction-following ability would make it more robust to such prompt-based supervision.

We trained the model using supervised causal language modeling (CLM), with masking applied to the instruction portion to ensure that only the review contributed to the loss. The model was trained for 3 epochs with an effective batch size of 16 (batch size 4 per device × 4 gradient accumulation steps). Training used the AdamW optimizer with a learning rate of 0.0002 across 26000 steps, and mixed precision training (fp16) was enabled to conserve memory.

Baseline Inference with Pre-trained Models

Our first experimental setup evaluated zero-shot generation using pre-trained GPT and LLaMA-2 models. Two prompt styles were used in all experiments:

1) "Give me a [positive/negative] review for the movie {movie_name}."
2) "Give me a [positive/negative] review for the movie {movie_name}. This is a short summary of the movie: {summary}."

Reviews generated from these prompts were evaluated against the original IMDb reviews using BLEU, ROUGE-L, semantic similarity (BERTScore), and sentiment match accuracy. These results established baseline performance before fine-tuning.

Fine-Tuned Model Evaluation

In our second experiment, we repeated the prompt-based inference pipeline using our fine-tuned GPT and LLaMA-2 models. These evaluations were conducted on 200 samples for GPT and 30 samples for the LLaMA-2 model, each using both the curated test set and training set, enabling analysis of generalization and memorization effects. We observed significant performance improvements in both lexical and semantic metrics as well as sentiment accuracy. For instance, fine-tuned GPT with the label-moviename prompt on the test set achieved:

Both prompt styles were tested for each fine-tuned model across both sets. The same evaluation pipeline—BLEU, ROUGE-L, BERT-based semantic similarity, and sentiment match accuracy—was used to compare generated reviews against reference reviews from the dataset.

Due to computational limitations and time constraints, the LLaMA-2 model, fine-tuned using QLoRA, was only evaluated on 30 samples. Despite using a quantized model, inference time on an NVIDIA L4 GPU remained substantial, primarily because of the model's size and adapter-based decoding overhead. This necessitated a significantly reduced evaluation set to maintain feasible experimentation timelines.

Controlled Decoding Experiments

For the final experiment, we studied how decoding parameters affect review quality and alignment of sentiment-conditioned reviews. We varied temperature, top-k, and top-p independently while keeping the other parameters fixed, and measured their influence on fluency, diversity, sentiment alignment, and semantic similarity upon prompting for These tests were conducted on the fine-tuned models using the test set and both prompt styles.

3.1. Objectives and Technical Challenges

This project aimed to explore sentiment-conditioned movie review generation by fine-tuning and evaluating LLMs, specifically GPT-2 and LLaMA-2 7B, on a curated IMDb dataset. The focus was on understanding how prompt structure and decoding strategies affect the fluency, sentiment alignment, and semantic quality of generated content. The models were trained using slightly different styles and objectives: GPT-2 was fine-tuned on instruction-summary-review triplets with explicit movie context, while LLaMA-2 was trained using generalized sentiment-based prompts without movie names or summaries.

Our primary objectives were to:

1. Curate, preprocess, and clean a dataset on movie names, summaries, and reviews based on the IMDb dataset from HuggingFace.
2. Construct a fine-tuning dataset for GPT-2 by extracting movie names and generating plot summaries using the OpenAI GPT-4 API, followed by cleaning and validation.
3. Fine-tune two models or slightly different objectives: GPT-2 on instruction-summary-review examples with prompt masking, LLaMA-2 7B using QLoRA on open-ended sentiment prompts and corresponding reviews.
4. Conduct controlled decoding experiments by generating reviews using varied decoding parameters—top-k, top-p, and temperature to study their effects on output diversity and sentiment control.
5. Assess model performance using a mix of quantitative metrics (BLEU, ROUGE-L, cosine similarity, sentiment classifier accuracy) and qualitative analysis of generated outputs.

Something we had to be mindful of was the potential for prompt overfitting, particularly with GPT-2. Fixed-format

prompts risked causing the model to memorize structure instead of learning sentiment. To reduce this, we used multiple paraphrased templates to increase prompt variation and encourage generalization.

Evaluating generative outputs posed another challenge. Metrics like BLEU and ROUGE penalize valid, creative responses that differ from the reference. To address this, we included semantic similarity (using Sentence-BERT) and sentiment classification (using a pre-trained BERT model) to better assess alignment and quality.

Fine-tuning LLaMA-2 7B required careful resource management. Initial attempts using PEFT caused out-of-memory errors on an NVIDIA L4 GPU. We resolved this by using 4-bit QLoRA, reducing batch size, and accumulating gradients. **Even so, LLaMA-2 training took ~24 hours, compared to ~7 hours for GPT-2.**

Inference with LLaMA-2 was slow and constrained. We were limited to a batch size of 2, and generating just 30 samples across one decoding setup (top-k, top-p, temperature) took ~**20 minutes**. A full parameter sweep was infeasible, so we tested only extreme values to capture behavioral differences.

The non-trivial dataset curation for our instruction fine-tuning task also added overhead. Extracting movie names and generating summaries using GPT-4 took ~2 full days (one each for train and test sets). We also had to clean vague or invalid outputs using regex filters and remove low-confidence samples to ensure data quality, which required both automated and manual efforts.

3.2. Problem Formulation and Design Description

The overall system design can be broken down into the following steps:

1) Generate reviews with GPT-2 and LLaMA-2 without fine-tuning them
2) Evaluate their responses using various metrics, like BERT scoring and BLEU scores
3) Fine-tune both decoders using summaries
4) Evaluate fine-tuned models' reviews in the same manner and compare

Learning Objective

Our problem can be formulated as a conditional text generation task and solved using causal language modeling with instruction prompts.

Given an input sequence $x = (x_1, x_2 \ldots x_n)$, consisting of a sentiment prefix and a movie summary, and a target sequence $y = (y_1, y_2 \ldots y_m)$, representing the desired review, the training objective is to minimize the negative log-likelihood of the target tokens given the input:

$$\mathcal{L}_{\text{CLM}} = -\sum_{t=1}^{m} \log P_\theta(y_t \mid y_{<t}, x)$$

Here, $P_\theta(y_t | y < t, x)$ denotes the probability of a token y_t, conditioned on all previous tokens $y_{<t}$ and the input x, computed by the autoregressive decoder with parameters θ. GPT-2 and LLaMA-2 both optimize this loss, but their training strategies and parameter update mechanisms differ.

Sampling Strategies for Decoding

Once the models are trained (or even when using pretrained models), output generation is performed via stochastic sampling from the predicted token distribution $P_\theta(y_t | y < t, x)$. Several strategies can be used to control the quality and diversity of generated text:

Top-k Sampling: At each time step ttt, only the top k most probable tokens are considered. The distribution is truncated and re-normalized:

$$P_k(y_t \mid y_{<t}) = \frac{P(y_t \mid y_{<t}) \cdot \mathbb{I}[y_t \in \text{TopK}_t(k)]}{\sum_{j \in \text{TopK}_t(k)} P(j \mid y_{<t})}$$

Top-p (Nucleus) Sampling: Instead of a fixed k, the top tokens whose cumulative probability exceeds a threshold p are selected:

$$\text{TopP}_t(p) = \{y_i : \sum_{j=1}^{i} P(y_j \mid y_{<t}) \leq p\}$$

Temperature Scaling: This scales the logits before softmax to control randomness. For temperature τ >0, logits z are adjusted as:

$$P(y_t) = \frac{\exp(z_t/\tau)}{\sum_j \exp(z_j/\tau)}$$

τ < 1 results in sharper distributions (more deterministic), whereas τ > 1 leads to flatter distributions (more diverse).

BERT Sentiment Classification/Evaluation

In our project, BERT plays a critical role as an evaluation model within the two-stage review generation pipeline. While GPT-2 and LLaMA-2 are used to generate movie reviews conditioned on prompt inputs, BERT served as a post-generation evaluator to assess whether the outputs align with the intended sentiment and maintain semantic coherence. We specifically used a pre-trained BERT model that was already fine-tuned for binary sentiment classification on the IMDB dataset from HuggingFace [9]. This fine-tuned classifier enabled us to quantitatively measure how well the generated reviews from both models adhered to the expected sentiment. The validity of the pretrained BERT model itself was lightly experimented on, and the metrics of this experiment are shown in the results section. These results verify that this model can perform sentiment classification for our task.

To summarize, we choose to avoid fine-tuning our own BERT binary sentiment classifier due to the following reasons:

1) There exists a positive-negative binary classification BERT model already
2) This model is trained on the same training set that we have available.
3) Fine-tuning our BERT would likely yield similar results to the off-the-shelf model. Our efforts, given the time constraints have gone towards exploring the decoder outputs. However, we leave this step as potential future work.

The decoder output evaluation process involved passing each generated review through the BERT classifier and comparing the predicted sentiment label to the intended label provided during generation. This provided a systematic way to detect cases where the generative models deviated from the expected sentiment or produced ambiguous or off-topic content. In addition to sentiment classification, we explored BERT's internal embeddings to perform semantic similarity checks between the generated review and reference reviews from the dataset, further assessing the relevance and coherence of the generated text.

The integration of BERT into our pipeline illustrates the use case of combining encoder-based discriminative models with decoder-based generative models. While generative models excel at producing fluent and diverse text, their outputs can be unreliable without supervision. BERT's ability to reliably classify and interpret text provides an effective safeguard and allows for a feedback loop that enhances the quality and trustworthiness of the overall system.

Evaluation Metrics Implementation

While our BERT pre-trained classifier plays a key role in evaluating sentiment alignment and semantic similarity, we additionally compute BLEU and ROUGE-L scores to assess the lexical overlap and surface-level fidelity between the generated reviews and the reference texts. Overall, our evaluation pipeline involves computing the following metrics:

BLEU Score (Bilingual Evaluation Understudy): For each generated review, we compute a sentence-level BLEU score against the reference review using NLTK. This metric captures n-gram precision and is smoothed to avoid zero scores for short sequences.

ROUGE-L Score (Recall-Oriented Understudy for Gisting Evaluation): This reflects the longest common subsequence between generated and reference reviews, emphasizing content recall and sequence coherence.

Semantic Similarity (Cosine Similarity of BERT Embeddings): Both reference and generated reviews are encoded using the [CLS] token embedding from a BERT model. Cosine similarity between corresponding pairs quantifies semantic alignment:

$$\text{cos\_sim}(x, y) = \frac{x \cdot y}{\|x\|\|y\|}$$

Sentiment Match Accuracy (via BERT Classifier): Generated reviews are passed through a pre-trained BERT sentiment classifier fine-tuned on IMDB. We compute the fraction of generated samples whose predicted sentiment matches the intended label from the prompt:
Sentiment Accuracy = # correct samples / total samples

Prompt Masking and Its Impact on the Objective Function

We observe divergence in the way both our models are fine-tuned, not just based on input data curation, but also in how loss is computed over the input-output pair.

Fine-tuning GPT-2 employs loss masking to exclude the prompt from contributing to the loss. If the input sequence is a concatenation of the tokens input = [PROMPT] || [REVIEW], we compute loss only over the [REVIEW] tokens. This ensures that the model focuses learning on how to generate sentiment-aligned reviews, not on reconstructing the prompt. This setup aligns with instruction-tuning practices, where only the response is supervised.

In contrast, for LLaMA-2 fine-tuning, we do not apply loss masking to the prompt. The model is trained to reconstruct the entire input sequence, which includes both the prompt and the target review. This means LLaMA-2 is partially incentivized to memorize or reconstruct the prompt, diluting its capacity to focus entirely on review generation.

4. Implementation

1. Data Collection and Preprocessing
2. Training and Evaluation Network

4.1 Data Collection and Preprocessing

The dataset used for sentiment classification in this project is the IMDb movie review dataset. The dataset consists of 50,000 movie reviews collected from the Internet Movie Database (IMDb) website, evenly split between positive and negative labels. The reviews are pre-divided into 25,000 training samples and 25,000 testing samples, with no overlap between the sets to ensure fair evaluation. The text lengths vary significantly, providing a diverse range of linguistic styles, vocabulary, and review structures. This dataset has become a standard for binary sentiment classification tasks due to its large size and high-quality human-labeled annotations. In our project, we used this dataset to evaluate the BERT-based sentiment classifier and to provide ground truth for training the decoder models.

We began with the original dataset, consisting of two columns: label (0 for negative, 1 for positive) and movie_review. To enrich the dataset with additional context, we used the OpenAI API to extract the movie name and generate a concise summary for each review. Extensive preprocessing was conducted to filter out samples where the movie name or summary could not be extracted. The final curated datasets included four columns: label, movie_name, summary, and movie_review, used separately for training and testing.

4.2 Training and Evaluation Network

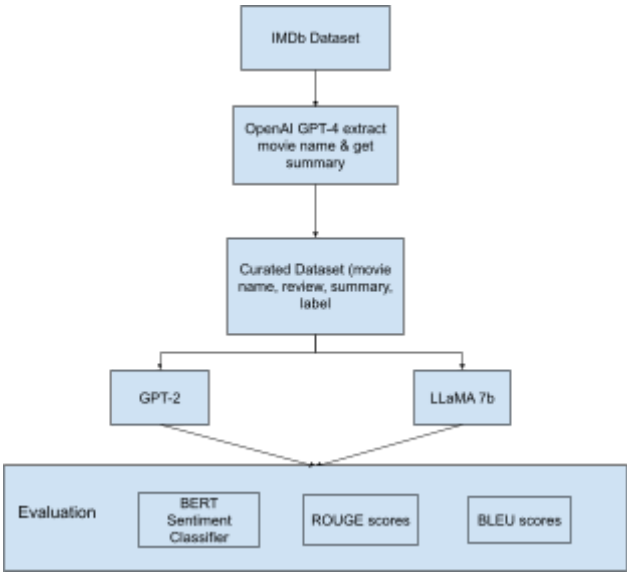

Figure 4: Entire Deep Learning Pipeline

4.3 Technical Stack

We integrated the OpenAI GPT-4 API to extract movie names and generate summaries during dataset curation, based on Hugging Face's standard IMDb dataset on movie reviews for sentiment analysis.

Our implementation leveraged the Hugging Face Transformers and Datasets libraries for model training, tokenization, and dataset management. For GPT-2 fine-tuning, we used the Trainer API along with a custom instruction masking strategy to ensure gradients were only propagated through the review portion of the input. Training loss was logged and visualized using TensorBoard for LLaMA-2, and was computed in the standalone script using TrainorCallbacks for GPT-2. Evaluation metrics such as BLEU, ROUGE, semantic similarity (Sentence-BERT cosine distance), and sentiment alignment (via a BERT classifier) were computed using evaluate, nltk, and torch.

For LLaMA-2 7B, we employed PEFT (Parameter-Efficient Fine-Tuning) using the QLoRA strategy. The model was quantized to 4-bit precision using bitsandbytes with NF4 quantization and bfloat16 computation, allowing training on limited memory hardware. We attached LoRA adapters using peft, and fine-tuned the model using Hugging Face's Trainer, with gradient accumulation to simulate larger batch sizes. All inference utilities, including prompt-based generation, classification, and evaluation, were modularized for reuse.

5. Results

5.1 Finetuning Evaluation

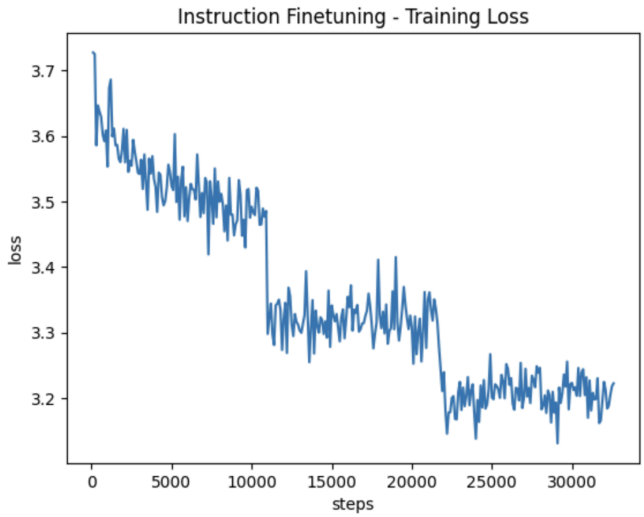

Figure 5: Training Loss for GPT-2

The GPT-2 instruction-tuned model demonstrated stable and effective convergence over approximately 32,000 training steps. The training loss decreased steadily from 3.7 to around 3.2, while the validation loss tracked closely behind, stabilizing near 3.44, indicating minimal overfitting and strong generalization. These results directly reflect key implementation choices in the training pipeline. The loss curve also exhibited sharp drops near 10k and 20k steps, indicating successful learning rate adjustments or optimizer restarts that contributed to convergence.

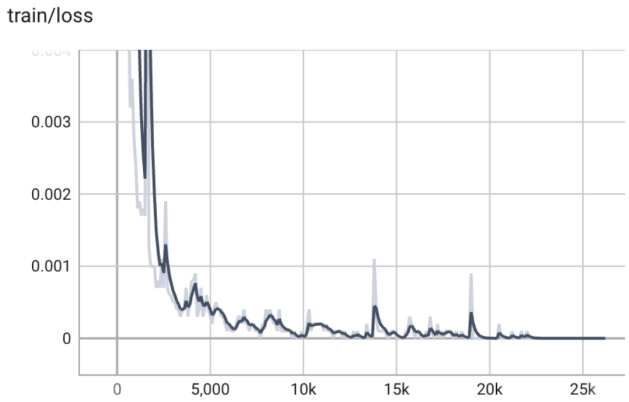

Figure 6: Loss curve for LLaMA-2

The training loss curve for LLaMA-2 fine-tuning shows a rapid decrease during the initial phase (0–5k steps), followed by a stable plateau with minor fluctuations. The final loss approaches zero, indicating effective convergence. Despite the near-zero training loss, LLaMA-2 retains generalization capability on unseen prompts as seen in our analysis below. This can be attributed to its extensive pretraining on diverse corpora, which provides strong inductive biases and robust language understanding. The LoRA head enables lightweight adaptation while keeping the pretrained backbone frozen, ensuring efficient learning with minimal risk of overfitting. The near-zero loss further suggests that the model converged quickly due to the small dataset size and strong prior knowledge; additional training steps beyond convergence likely had limited benefit.

Evaluation of Pretrained BERT Model

As was mentioned in the methodology of the project, a fine-tuned BERT from HuggingFace was used in our project. When choosing between fine-tuning BERT for ourselves and choosing an existing instance off-the-shelf, it was important to justify whether fine-tuning would offer any benefit about binary classification accuracy. To evaluate the fine-tuned BERT model, we performed inference on our sentiment-labeled training set and testing set. The model scored a 99.83 % on the training set and 89.09% on the testing set. This discrepancy can be attributed to the fact that the off-the-shelf model was trained on this exact training set, and thus has memorized the training set well

5.2 Pre-trained Model Inferencing Analysis

Qualitative Analysis

Movie: The Manson Family
Sentiment: negative
Generated Review:
Unterscheidung zwischen Kern und Schale, die nach dem Schalensatz zu unterscheiden sind, liegt meist in der Bezeichnung des Schalensatzes.
Negative review.
Negative reviews are those that are not positive.
Negative reviews are those that are not positive. A negative review can be a review that is either negative in nature or not positive in nature. A negative review is a review that is not positive. A negative review can be a review that is either negative in nature or not positive in nature. A negative review can be a review that is either negative in nature or not positive in nature. A negative review can be a review that is either negative in nature or not positive in nature. A negative review can be a review that is either negative in nature or not positive in nature. A negative review can be a review that is either negative in nature or not positive in nature.

Figure A: Generate a review from the Pre-trained LLaMA-2 model using sentiment + movie name only prompt style

Movie: The Manson Family
Sentiment: negative
Generated Review:
The movie is set to release in 2023.
The Manson Family movie is about a cult leader who manipulates his followers to commit murder.
The Manson Family movie is about a cult leader who manipulates his followers to commit murder. The movie is based on the true story of Charles Manson and his followers, who committed a series of murders in the 1960s. The movie explores the events leading up to the murders, as well as the aftermath of the crimes. The Manson Family is a crime drama.

Figure B: Generate a review from the Pre-trained LLaMA-2 model using sentiment + movie name + summary prompt style

Analysis for Figure A:

1) Extreme repetition of definition-level phrases,
2) May boost BLEU/ROUGE because it uses known tokens repeatedly, but yields very low semantic content

Analysis for Figure B:

1) Reads like a factual synopsis, possibly from training data.
2) Shows zero sentiment despite prompts indicating "negative".
3) May get high semantic similarity, since the summary and generation are nearly identical, but fails as a review.

13. Movie: *Beware, My Lovely*
Sentiment: negative
Generated Review:
I just didn't want to get the movie. I'm not a fan of the story, I think it's horrible, and I think it's a bad movie. It's a terrible movie. It's a movie about love. It's a movie about a woman who's not only not a good person but she's also a bad person. I don't like it. I don't like it. I don't like the movie.
I'm not sure I want to see a movie like this. I'm not sure I want to see a movie like this. I'm not sure I want to see a movie like this. I'm not sure I want to see a movie like this. I'm not sure I want to see a movie like this.

Figure C: Generate a review from the Pre-trained GPT model using sentiment + movie name only prompt style

13. Movie: *Beware, My Lovely*
Sentiment: negative
Generated Review:
Her only hope is to gain a restraining order against her, but she is unable to get one, and so her friend, a young man named Tom, is sent to assassinate her. In the process, Tom learns that the war widows are in fact his own children, and he must find them in order to save their loved ones. Written by Tony Kornheiser kornheiser@hotmail.com

Figure D: Generate a review from the Pre-trained GPT model using sentiment + movie name + summary prompt style

Analysis for Figure C:

1) This review repeats the same sentiment without detail
2) The repetition inflates BLEU/ROUGE scores due to n-gram overlap, but adds no actual semantic value
3) This output likely gets high sentiment accuracy, but low semantic quality

Analysis for Figure D:

1) This review is a copy or hallucinated dump of a plot summary, potentially from pre-training data or web crawl noise.
2) It misses the sentiment completely. Fails both as a review and as a sentiment-controlled generation.

Quantitative Analysis

		Bleu	ROUGE	Semantic Similarity	Sentiment Match Accuracy
PRETRAINED GPT	label-movie name	0.0039	0.1202	0.2078	0.54
	labe-moviename-summary	0.0039	0.1202	0.2078	0.54
PRETRAINED LLAMA	label-movie name	0.0025	0.0892	0.1249	0.5333
	labe-moviename-summary	0.003	0.1082	0.1756	0.5667

Table 1: Metrics after performing inference on pre-trained GPT and LLaMA-2 (before fine-tuning)

Combined Analysis

Before applying any fine-tuning, we evaluated the pretrained GPT and LLaMA-2 models on sentiment-controlled movie review generation using both quantitative metrics and qualitative outputs. Quantitatively, both models showed very limited sentiment alignment, with sentiment match accuracy hovering around 54–57% and semantic similarity scores below 0.21 indicating weak task grounding. Prompt style had minimal effect at this stage, and BLEU/ROUGE remained low due to incoherent or repetitive output.

Qualitatively, the models failed to generate coherent reviews aligned with the given sentiment. The GPT

outputs either repeated shallow sentiment phrases (e.g., “I don’t like it”) or hallucinated content (especially when summaries were included). LLaMA-2 exhibited even stronger issues, including copying definitions or reverting to template-like phrasing. In both models, adding summaries often worsened the output by introducing sentiment-conflicting signals or distractor content. Overall, the analysis underscores that pretrained models lack sufficient instruction-following or conditioning alignment for this task and require fine-tuning to meaningfully respond to structured prompts.

5.3 Fine-tuned Model Inferencing Analysis

Hyperparameter Testing (Top-p, Top-k, Temperature)

To further understand how decoding configurations influence model behavior, we systematically varied top-p, top-k, and temperature settings across different prompt styles. These experiments reveal how sampling diversity interacts with prompt clarity, affecting both output quality and alignment with the intended sentiment:

Prompt Style	BLEU	ROUGE	Semantic Similarity	Sentimer Accuracy
Sentiment + Movie	0.0043	0.1296	**0.5808**	**0.9000**
Sentiment + Movie + Summary	**0.0000**	**0.0597**	**0.2325**	**0.6333**

Table 2: baseline for LLaMA-2’s sentiment-controlled generation under standard decoding(top-p = 0.95, top-k = 50, temperature = 0.7)

The Sentiment + Movie prompt significantly outperforms the summary augmented version across all metrics:

1) BLEU: 0.0043 → 0.0000 - Summary prompt output has near-zero n-gram overlap with the reference. This could be due to hallucination or off-topic phrasing
2) Semantic Similarity: 0.5805 → 0.2325 - Indicates a major loss in meaning alignment and sentiment coherence

Summaries often carry implicit sentiment, which may contradict the label and confuse the model.

top-p	Prompt Style	BLEU	ROUGE	Semantic Similarity	Sentiment Accuracy
0.4	Sentiment + Movie	0.0050	0.1213	0.5385	0.8667
0.4	Sentiment + Movie + Summary	0.0050	0.1355	0.4802	0.8333
0.7	Sentiment + Movie	0.0054	0.1258	0.5170	**0.9000**
0.7	Sentiment + Movie + Summary	**0.0072**	0.1351	**0.5705**	0.8667
0.95	Sentiment + Movie	0.0043	0.1296	**0.5808**	**0.9000**
0.95	Sentiment + Movie + Summary	0.0000	0.0597	0.2325	0.6333

Table 3: Varying top-p for LLaMA-2 on the test set

1) Low top-p (0.4): Sampling is tightly restricted to high-probability tokens
    a) An advantage of this would be that it prevents hallucinations and off-topic drift
    b) A disadvantage is that this limits adaptability, especially with complex prompts where summaries are included
    c) Semantic similarity and sentiment accuracy are moderate but plateau early
2) Moderate top-p (0.7): This value strikes the best balance
    a) For summary prompts, we achieve a high semantic similarity(0.5705), which indicates controlled rewording and sentiment preserved
    b) It allows the model to explore varied phrasing while staying task-aligned
3) High top-p (0.95): Introduced excessive diversity
    a) Effective for sentiment + movie name only prompts, achieves the highest semantic similarity (0.5808) due to its simplicity.
    b) This degrades the summary performance (semantic similarity drops to 0.2325) as the model samples less probable tokens that may conflict with the intended sentiment.

top-k	Prompt Style	BLEU	ROUGE	Semantic Similarity	Sentiment Accuracy
10	Sentiment + Movie	**0.0089**	**0.1417**	**0.6032**	**0.9333**
10	Sentiment + Movie + Summary	0.0043	0.1336	0.5090	0.8667
50	Sentiment + Movie	0.0043	0.1296	0.5808	0.9000
50	Sentiment + Movie + Summary	0.0000	0.0597	0.2325	0.6333
300	Sentiment + Movie	0.0067	0.1320	0.5631	0.9333
300	Sentiment + Movie + Summary	0.0063	0.1322	**lowest: 0.3977**	**lowest: 0.7667**

Table 4: Varying top-k for LLaMA-2 on the test set

1) Top-k=10 performs best across all metrics for both prompt styles:
    a) BLEU = 0.0089, Semantic Similarity = 0.6032, Semantic Accuracy = 0.9333
    b) The model remains tightly focused on high probability tokens, improving fluency and sentiment grounding
    c) This is particularly helpful for summary prompts, which benefit from output constraints to mitigate ambiguity
2) Top-k = 50 maintains strong performance on sentiment + movie only prompts, but causes a collapse for Summary prompts:
    a) Semantic similarity for summary drops to 0.2325 - the model begins confusing conditioning context with generation target
3) Top-k = 300 allows a lot of freedom in token selection
    a) This fails on summary-based inputs - semantic similarity drops to 0.3977, and sentiment accuracy to 0.7667
    b) This indicates conceptual drift: the model samples irrelevant tokens that dilute the intended tone

Temp	Prompt Style	BLEU	ROUGE	Semantic Similarity	Sentiment Accuracy
0.1	Sentiment + Movie	0.0039	0.1207	0.5278	0.9000
0.1	Sentiment + Movie + Summary	0.0045	0.1363	0.4471	0.8333
0.7	Sentiment + Movie	0.0043	0.1296	**0.5808**	0.9000
0.7	Sentiment + Movie + Summary	0.0000	0.0597	0.2325	0.6333
1.3	Sentiment + Movie	0.0051	0.1245	0.5618	**0.9333**
1.3	Sentiment + Movie + Summary	0.0043	0.1205	0.5028	0.8333

Table 5: Varying temperature for LLaMA-2 on the test set

1) Low Temperature (0.1):
    a) Keeps sentiment alignment strong (0.9000 accuracy), but limits fluency and variation
    b) Summary prompts suffer (semantic similarity = 0.4471) - lacking flexibility to handle an ambiguous structure
2) High Temperature (1.3):
    a) This works well for clean, short prompts - sentiment accuracy reaches 0.9333 with good semantic preservation (0.5618)
    b) But fails to meaningfully fix issues in prompts with summary - semantic similarity stays lower (0.5028).
3) Default temperature (0.7):
    a) Balances consistency and diversity, yielding strong semantic similarity (0.5808) and sentiment accuracy (0.9000) for sentiment + Movie only prompts
    b) But collapses on prompts with summaries - semantic similarity drops to 0.2325.

Prompt Style	BLEU	ROUGE	Semantic Similarity	Sentiment Accuracy
Sentiment + Movie Name	**0.0065**	**0.1377**	**0.6722**	**0.980**
Sentiment + Movie Name + Summary	0.0057	0.1359	0.6609	0.935

Table 6: Baseline for GPT's sentiment-controlled generation under standard decoding(**top-p = 0.95**, **top-k = 50**, **temperature = 0.7**)

1) Sentiment + Movie Name prompt outperforms all other setups
    a) BLEU 0.0065/ROUGE 0.1377: Strong n-gram and recall overall with reference review
    b) Semantic Similarity 0.6722: The generated review closely matches the intended meaning and sentiment
    c) Sentiment Accuracy 0.980: GPT nearly always picks the correct sentiment when the prompts are simple
2) Adding a summary to the prompt slightly degrades performance:
    a) Semantic similarity falls by ~1.7% (0.6722 → 0.6609)
    b) Semantic Accuracy drops to 0.935, showing occasional blending of summary cues with the instructed sentiment.

Summaries can carry their emotional language, which GPT may treat as a generation target rather than context, despite instruction tuning.

top-p	Prompt Style	BLEU	ROUGE	Semantic Similarity	Sentiment Accuracy
0.95	Sentiment + Movie Name	**0.0065**	**0.1377**	**0.6722**	**0.980**
0.95	Sentiment + Movie Name + Summary	0.0057	0.1359	0.6609	0.935
0.4	Sentiment + Movie Name	0.0033	0.1081	0.5650	0.940
0.4	Sentiment + Movie Name	0.0044	0.1240	0.6027	0.920
0.7	Sentiment + Movie Name	0.0051	0.1279	0.6332	0.965
0.7	Sentiment + Movie Name + Summary	0.0059	**0.1380**	**0.6546**	0.955

Table 7: Varying top-p for GPT on the test set

1) Top-p = 0.95
    a) Best performance across metrics for sentiment + movie only prompts: BLEU = 0.0065, Semantic Similarity = 0.6722, Sentiment Accuracy = 0.980
    b) Adding a summary to the prompt keeps the performance strong (0.935 accuracy), showing GPT's robustness to mixed inputs.
2) Top-p = 0.4 (Low diversity)
    a) Outputs become rigid (BLEU = 0.0033, Similarity = 0.5650)
    b) Sentiment accuracy (0.940) is maintained
    c) This means that the limited token pool hinders response quality, especially with summaries.
3) Top-p = 0.7 (Moderate Diversity):
    a) Best balance for both prompt styles: prompts with summaries achieve 0.6546 semantic similarity and 0.955 sentiment accuracy.
    b) Moderate diversity allows GPT to rephrase while staying aligned to the prompt intent.

top-k	Prompt Style	BLEU	ROUGE	Semantic Similarity	Sentiment Accuracy
50	Sentiment + Movie Name	**0.0065**	0.1377	**0.6722**	0.980
50	Sentiment + Movie Name + Summary	0.0057	0.1359	0.6609	0.935
10	Sentiment + Movie Name	0.0067	**0.1389**	0.6666	0.950
10	Sentiment + Movie Name + Summary	0.0062	**0.1393**	0.6559	0.925
300	Sentiment + Movie Name	0.0065	0.1371	0.6699	**0.985**
300	Sentiment + Movie Name + Summary	**0.0070**	0.1365	**0.6303**	**0.915**

Table 8: Varying top-k for GPT on the test set

1) Top-k = 50 (moderate diversity)
    a) Strong overall performance, especially for sentiment + movie name only prompts (semantic similarity = 0.6722, sentiment accuracy = 0.980)
    b) Balances expressive generation with task alignment
2) Top-k = 10 (low diversity)
    a) Slight gains in BLEU and ROUGE from deterministic completions
    b) A small drop in semantic similarity suggests more rigid outputs with reduced paraphrasing flexibility
3) Top-k = 300 (High diversity)
    a) Achieves highest sentiment accuracy (0.985) for sentiment + movie only

prompts, proving GPT's fluency under wide token sampling

b) But performance degrades with prompts with summary (Similarity = 0.6303, Accuracy = 0.915) due to over-sampling from low-probability tokens

Temp	Prompt Style	BLEU	ROUGE	Semantic Similarity	Sentiment Accuracy
0.7	Sentiment + Movie Name	**0.0065**	**0.1377**	**0.6722**	**0.980**
0.7	Sentiment + Movie Name + Summary	0.0057	0.1359	0.6609	0.935
0.1	Sentiment + Movie Name	0.0026	0.0915	0.4497	0.900
0.1	Sentiment + Movie Name + Summary	0.0037	0.1198	0.5605	0.905
1.3	Sentiment + Movie Name	0.0049	0.1232	0.5843	0.855
1.3	Sentiment + Movie Name + Summary	0.0042	0.1185	0.5678	0.865

Table 9: Varying Temperature for GPT on the test set

1) Default temp = 0.7
    a) Has the best overall performance
    b) Semantic Similarity = 0.6722, Sentiment Accuracy = 0.980 for the sentiment + movie only prompts
    c) Even with the summaries added to the prompt, it maintains strong alignment (Similarity = 0.6609, Accuracy = 0.935)
2) Low Temp = 0.1
    a) Output becomes rigid and template-like
    b) BLEU = 0.0026, Semantic similarity = 0.4497 → low variation, underuse of context
3) High Temp = 1.3
    a) Increases diversity (Semantic similarity = 0.5843), but reduces sentiment control (accuracy = 0.855)

Qualitative Analysis

We conducted a series of qualitative experiments aimed at closely examining the linguistic and semantic characteristics of the generated reviews. These examples provide deeper insight into model behavior: highlighting issues like repetition, hallucination that numerical metrics alone may fail to capture**.**

2. Movie: *Night of the Demons*
Sentiment: negative
Generated Review:
I have watched it twice, both times in succession and on the first time I was watching, I wanted to think I could get my money back, and at the end I did. My friends and I went to see the DVD and both times we both agreed to watch it again. The second time we both agreed it was a bit better. I'm sorry, but I've been a fan of the show since I was a child, and this movie is the worst movie ever! I watched it because I was watching the "new" movie, The Ring, and I thought I'd give it a chance. I've seen this movie twice, both times in succession, and it's not even entertaining anymore! It's a waste of my time and money. This movie is not worth watching. I've seen this movie twice, both times in succession. The second time I had

Figure E: Generate a review from the fine-tuned GPT model using sentiment + movie name only prompt style

2. Movie: *Night of the Demons*
Sentiment: negative
Generated Review:
This movie is a classic. I am a huge fan of the horror genre and have watched this movie many times. So many movies have been made where a movie fails in every way, and this one is no exception. If you have not seen it, I strongly suggest you do not. This movie is just plain horrible. Not only is it a bad movie, but it is also a horrible movie with no plot. The acting is not even close to what you would expect from a horror movie. The music is very bad. The characters are just plain bad. I would not recommend this movie to anyone. The only reason I did not give it a 1

Figure F: Generate a review from the fine-tuned GPT model using sentiment + movie name + summary only prompt style

Analysis for Figure E:

1) Shows personal experience, justification, and clarity in negativity.
2) Criticisms are concrete (e.g., "acting is bad", "waste of time"), which likely boosts both semantic similarity and sentiment classification.
3) No hallucinated summary or repetition.

Analysis for Figure F:

1) Still clearly negative, but now reflects more of the summary's factual elements.
2) Summaries have off-task content that GPT doesn't know whether to reuse or challenge. Some sentences repeat summary phrasing.

6. Movie: *The Beguiled*
Sentiment: positive
Generated Review:
I thought this was an excellent film. I was expecting it to be somewhat lame, but I was pleasantly surprised. The characters were believable and the acting was superb. I thought Nicole Kidman, who I've never been a fan of, was very good. I also thought Colleen Camp and the young girl who played Alicia were fantastic. I've read the book and I must say that I think they did a better job of adapting the book than the Coen brothers did of *O, Brother, Where Art Thou?* The only thing I didn't like about the movie was the ending. I thought it was a little too abrupt. It didn't really wrap things up. Other than that, I thought the movie was wonderful. I recommend it. It's definitely worth watching. 8/10. – MERK

Figure G: Generate a review from the fine-tuned LLaMA-2 model using sentiment + movie name only prompt style

6. Movie: *The Beguiled*
Sentiment: positive
Generated Review:
The movie begins with the young girls in a boarding school in the middle of the war. A Union soldier is brought in after being wounded in battle. The headmistress, played by Geraldine Fitzgerald, is a very religious woman who is strict and harsh with the girls. She is not a very good teacher. She is a woman of the past. She is the typical woman who is obsessed with men, and she is obsessed with this soldier, who she considers "God's gift to women." She is so obsessed with him that she is willing to do anything to keep him there.

Figure H: Generate a review from the fine-tuned LLaMA-2 model using sentiment + movie name + summary prompt style

Analysis for Figure G:

1) Fully follows sentiment + gives justification.
2) Offers natural phrasing, good transitions, and no repetition.
3) LLaMA-2 shows strong improvement over pretrained, though the tone is still slightly generic.

Analysis for Figure H:

1) This is again a restatement of the summary with very faint sentiment cues.

Quantitative Analysis

To assess the models' performance more systematically, we conducted quantitative experiments measuring BLEU, ROUGE, semantic similarity, and sentiment accuracy across both training and test sets. The following tables summarize how GPT and LLaMA-2 respond to different prompt styles under default decoding settings.

top-k/top-p /temperature config	Data Set Used	PROMPT STYLE	Bleu	Rouge	Semantic Similarity	Sentiment Match Accuracy

DEFAULT
 	
`temperature=0.7`
 	
`top_k=50`
 	
`top_p=0.95`
	Train	Sentiment + Movie Name	0.009	0.1349	0.6573	0.9667

		Sentiment + Movie Name + Summary	0.0068	0.1343	0.5551	0.9667
	Test	Sentiment + Movie Name	0.0043	0.1296	0.5808	0.9
		Sentiment + Movie Name + Summary	0.0	0.0597	0.2325	0.6333

Table 10: Metrics of inferencing on the finetuned LLaMA-2 model on the train and test curated datasets

The fine-tuned LLaMA-2 model performs well on the train set when prompted with just sentiment and movie name, it achieves high semantic similarity (0.6573) and perfect sentiment accuracy (0.9667), showing strong alignment and fluency. However, adding a summary reduces semantic similarity (to 0.5551), indicating that LLaMA-2 struggles to balance richer input without losing focus, though sentiment accuracy remains stable during training.

On the test set, this weakness becomes more severe. Performance with the simple prompt is still decent (semantic similarity 0.5808, accuracy 0.9), but the model fails with the summary prompt: BLEU drops to 0.0, semantic similarity to 0.2325, and sentiment accuracy to just 63.3%. This shows that LLaMA-2 often misinterprets sentiment when input complexity increases.

Overall, LLaMA-2 is effective with clean, structured prompts but breaks down with ambiguous or

sentiment-rich summaries, due to its inability to separate instruction from context during generation.

top-k/top-p /temperature config	Data Set Used	PROMPT STYLE	Bleu	Rouge	Semantic Similarity	Sentiment Match Accuracy
DEFAULT temperature=0.7 top_k=50 top_p=0.95	Train	Sentiment + Movie Name	0.006	0.1397	0.7883	0.97
		Sentiment + Movie Name + Summary	0.0019	0.1025	0.6191	0.855
	Test	Sentiment + Movie Name	0.0065	0.1377	0.6722	0.98
		Sentiment + Movie Name + Summary	0.0057	0.1359	0.6609	0.935

Table 11: Metrics of inferencing on the fine-tuned GPT model on the train and test curated datasets

The fine-tuned GPT model shows strong performance across both train and test sets, especially with the simpler "Sentiment + Movie Name" prompt. It achieves the highest sentiment accuracy (0.97 train, 0.98 test) and strong semantic similarity (0.7883 train, 0.6722 test), confirming that GPT reliably follows explicit instructions.

When summaries are added, performance drops, and semantic similarity and sentiment accuracy both decrease slightly (e.g., 0.6609 and 0.935 on the test set). This suggests GPT sometimes incorporates emotional cues from the summary that may conflict with the instructed sentiment.

Unlike LLaMA-2, GPT remains robust even with richer prompts, due to its instruction tuning. However, the drop in BLEU and sentiment accuracy on the training set (to 0.0019 and 0.855) shows that even GPT may blur context and instruction if summaries are too sentiment-driven

Combined Analysis

Combining **quantitative metrics** with **qualitative analysis** reveals a key mismatch between numerical scores and actual model behavior. Metrics like BLEU, ROUGE, and sentiment accuracy reward surface-level traits such as token overlap or sentiment keywords but often miss coherence, content quality, and task completion. For example, a model may score high by repeating "I don't like it," yet fail to deliver a meaningful review. Qualitative inspection shows these outputs are metric-compliant but semantically hollow, highlighting the limits of relying solely on automated scores for evaluating sentiment-controlled generation.

Clean prompts like "Sentiment + Movie Name" consistently produce more grounded, coherent, and review-like outputs in both GPT and LLaMA-2. These prompts reduce confusion by encouraging generation from scratch and avoiding sentiment leakage or hallucinations. In contrast, summary-augmented prompts introduce ambiguity, embedding conflicting sentiment or tempting the model to echo summary language, leading to misalignment despite decent metrics. Together, the results emphasize the need for hybrid evaluation: true performance emerges only when quantitative scores are paired with qualitative insights.

Ultimately, the best-performing setup across both models was the combination of the "Sentiment + Movie Name" prompt style with moderate sampling parameters: specifically top-p = 0.95, top-k = 50, and temperature = 0.7. This configuration consistently achieved the highest sentiment accuracy and semantic similarity while producing fluent, review-like text. From a qualitative standpoint, this makes sense: the prompt is clean and unambiguous, giving the model clear instructions, while the sampling diversity is balanced, allowing for expressive but grounded generation without drifting into repetition or hallucination.

5.4 Discussion and Insights Gained

Dataset Curation

A major insight from this project was the importance of dataset preprocessing and curation. Our initial IMDb dataset contained only labels and reviews. We augmented this by using an OpenAI key to extract movie names and summaries for each review. This step required substantial effort, as we had to filter out noisy samples where a movie name could not be reliably extracted or a meaningful summary could not be generated. Careful dataset cleaning ensured that both our fine-tuning and evaluation experiments were grounded in consistent, high-quality input data, which greatly influenced downstream model performance. Thus, a major insight relates to the importance of the dataset when training a model, and that it is worth it to spend time curating a higher-quality dataset if resources are available.

GPT-Specific Fine-tuning

In our initial experiments with GPT-2, we fine-tuned the model directly on causal language modeling loss without masking, using a batch size of 4. This led to severe failure cases where the model either echoed parts of the prompt or simply restated the instruction with minimal variation. Typical outputs included repeated tokens such as "Give me a negative review review review..." or trivial responses like "This is a positive review." We realized that without masking out the instruction portion during loss calculation, the model was wasting gradient updates on tokens it had already been given. Our key takeaway was that "loss is only meaningful when it is targeted." Masking the instruction and only applying loss to the review portion provided much clearer gradient signals and helped the model focus on the intended task.

We then applied instruction fine-tuning to GPT-2, using structured prompts like “Give me a positive review for the movie {movie name}. Summary: {summary}”. We also experimented with two variations: one with just the movie name and label, and another including both the movie name and a short movie summary. Reducing batch size increased the number of gradient updates per example, which improved stability. We found that this design, combined with masking and clear task-aligned prompting, significantly improved both the fluency and task alignment of GPT-2's outputs. The model became much better at following instructions and understanding the domain of movie reviews with clear sentiment targets.

LLaMA-2-Specific Fine-tuning

For LLaMA-2 7B, we used QLoRA to fine-tune the model while freezing the full pretrained weights and only updating low-rank adapter layers. Remarkably, we trained LLaMA-2 using the same dataset that had caused the initial GPT-2 failure. LLaMA-2 quickly adapted to the instruction-following task. We attribute this to both the larger model size and possibly a richer pre-training set that exposed LLaMA-2 to instruction-style data. Using QLoRA allowed us to efficiently adapt a large model with minimal memory requirements and without degrading its strong pretrained language fluency. We also observed that LLaMA-2 needed very little explicit prompt engineering compared to GPT-2.

General Fine-tuning

Our experiments also explored the effects of varying inference parameters like top-k, top-p, and temperature. We ran controlled experiments on both fine-tuned GPT-2 and LLaMA-2 models using our curated test set. We held two parameters constant while systematically varying the third to observe its effect on generation style. We found that lowering temperature produced more deterministic and conservative outputs, while increasing temperature led to more diverse but occasionally less coherent generations. These experiments highlighted the trade-off between diversity and consistency when using sampling-based generation at inference time.

Lastly, we compared performance between pretrained models and fine-tuned models using automatic evaluation metrics. In the pretrained phase, sentiment match accuracy was very low (around 54%), with poor BLEU and semantic similarity scores. After fine-tuning, both models improved drastically. Fine-tuned GPT-2 achieved a 97% sentiment match accuracy and much higher BLEU, ROUGE-L, and semantic similarity scores, showing the clear impact of instruction fine-tuning and dataset alignment. We replicated this exact experiment structure on fine-tuned LLaMA-2 and observed comparable gains. This confirmed the effectiveness of both instruction fine-tuning for smaller models and adapter-based fine-tuning for larger models under constrained compute budgets.

6. Future Work

Additional Decoder Models

While this study provides a foundational comparison between pre-trained and fine-tuned versions of GPT-2 and LLaMA-2 models, there remains a substantial opportunity for further exploration. One possibility would be to try more models. For instance, newer and larger-scale models such as GPT-3, GPT-4, and larger LLaMA-2 models could be included, with access to the appropriate computing resources, to assess how improved architectures might impact movie review generation. Additionally, future efforts could evaluate the impact of alternative fine-tuning approaches, such as low-rank adaptation (LoRA), reinforcement learning with human feedback (RLHF), or parameter-efficient fine-tuning (PEFT) methods [6, 7, 8].

Testing Process

Our evaluation relied on BLEU, ROUGE-L, cosine similarity (Sentence-BERT), and sentiment match accuracy. To enhance this framework, future work could incorporate more targeted metrics. BERTScore offers a deeper measure of semantic similarity. MAUVE can assess distributional alignment between human and model-generated text. Perplexity, computed using an external language model, can quantify fluency.

We also suggest adding LLM-based evaluators (e.g., GPT-4) for judging factual consistency between summaries and reviews. Natural Language Inference

(NLI) models could test logical entailment between input and output.

Robustness could be tested using adversarial prompts, ambiguous sentiment cues, or non-English inputs to probe model generalization and reliability.

Fine-tuning BERT Classifier Ourselves

In the methodology section, we deemed it appropriate to use an off-the-shelf BERT model as our binary classifier, with the reasoning that our efforts would be better spent improving upon the decoder outputs [9]. However, the original off-the-shelf BERT was fine-tuned with the following parameters:

1) 5 epochs
2) Learning rate of 2e-5
3) Maximum sequence length of 128

Experiments could be performed in the future to properly test these parameters and get more optimal values, if they exist, and could help our binary classifier perform better.

Real vs. Synthetic Reviews

During our experiments, we used a fine-tuned BERT model to classify binary sentiments and compare the semantic meaning of decoder outputs and real, ground-truth reviews. Another compelling direction for future research is to fine-tune BERT to distinguish between human-written and LLM-generated movie reviews. As generative models become increasingly fluent and capable, it has become harder to tell whether a given text is "fake" or "real" (Did it come from humans or machines?). Detecting the origin of a review has significant implications for trust, transparency, and responsible AI deployment, and could be a potential next step in the natural course of this project.

To pursue this, a dataset of labeled examples containing both human-written and model-generated movie reviews would first need to be curated. To keep this project self-contained, the decoder models we trained could be used to generate those synthetic reviews. The dataset could then be used to fine-tune BERT with a binary classification head to learn patterns that differentiate between the two classes. We hypothesize that such patterns might include subtle stylistic inconsistencies, unusual phrasing, repeating patterns, or a lack of personal detail often found in generated content. The challenge lies in the fact that current LLMs can produce text nearly indistinguishable from human writing, making this a difficult task.

Successfully fine-tuning BERT in that way would not only help in identifying machine-generated reviews but could also help in some sort of decoder feedback loop. It could be used to filter out lower-quality or less authentic outputs after an initial generation, and therefore increase the credibility of our generative models in writing synthetic reviews (that are indistinguishable from real reviews).

Increasing Sentiment Range

Finally, another path for future exploration lies in expanding the review generation framework from binary sentiment classification (positive/negative) to more fine-grained outputs, such as generating reviews corresponding to 1-to-5 star ratings. This would make the system more aligned with how real-world movie reviews are structured on platforms like IMDb and Rotten Tomatoes, where nuanced distinctions between "good" and "excellent" or "poor" and "terrible" reviews are meaningful to users. Achieving this would require significant modifications to both the dataset and model conditioning strategies. Specifically, fine-tuning datasets would need to be relabeled or restructured to include discrete star ratings rather than binary sentiment tags. Additionally, new prompt templates or control tokens would be necessary to guide the generative models (GPT-2 and LLaMA-2) to produce outputs reflective of each star level.

The model architectures themselves are well-suited to handle this change, but the training process must be carefully adjusted to avoid ambiguity between adjacent classes (e.g., distinguishing a 3-star from a 4-star review). A multi-class classification head could also be added to BERT or a similar model for evaluation purposes, allowing the system to automatically assess whether the generated review accurately reflects the intended rating. This would represent a more challenging evaluation task compared to simple positive/negative sentiment, but it would also offer a much richer and more practical application.

This direction is particularly interesting because it opens the possibility of further generalizing the system to other nuanced sentiment scales or even subjective categories such as "action-packed," "family-friendly," or "critically acclaimed." Moving toward controlled, multi-dimensional text generation would make the system more valuable for real-world recommendation systems and personalized content generation. It also provides a meaningful framework for exploring how large language models handle subtle distinctions in sentiment, tone, and opinion. This proposed extension would represent a natural and impactful evolution of the work started in this project.

7. Conclusion

This project explored the task of sentiment-controlled movie review generation using GPT and LLaMA-2. By curating a structured dataset of movie names, summaries, and reviews, and implementing a masked loss fine-tuning strategy, we aimed to condition generations on explicit sentiment and metadata. The primary objective was to generate fluent, sentiment-aligned reviews that remain faithful to the prompt intent.

Through extensive evaluation combining quantitative metrics (BLEU, ROUGE, semantic similarity, sentiment accuracy) and qualitative inspection of generated outputs, we found that clean prompt formats, particularly the "Sentiment + Movie Name" style, outperformed summary-augmented prompts in both accuracy and coherence. Fine-tuning significantly enhanced generation quality across models. Match accuracy improved from 54% (just pre-trained) to 98% (post fine-tuning) in GPT and from 53% (just pre-trained) to 96.7% (post fine-tuning) in LLaMA-2, along with substantial gains in semantic similarity. These improvements were achieved by moving beyond pre-training alone, where both models initially struggled to follow sentiment conditioning or produce coherent reviews. The results demonstrate that pre-training is not sufficient for this task; fine-tuning with task-specific objectives and prompt-aware loss strategies is essential for aligning model outputs with the desired sentiment and review structure. Thus, the original goals of the project were successfully met.

A key lesson learned is that automatic metrics alone are not sufficient to evaluate controllable generation. High BLEU or sentiment accuracy can obscure semantic drift, hallucinations, or prompt leakage. Qualitative analysis revealed that models often exploit shortcuts like repetition or summary echoing that inflate metrics without producing meaningful reviews. As a result, we emphasize the importance of hybrid evaluation approaches combining numerical scores with human interpretability.

Looking forward, future work can focus on improving model robustness to noisy inputs like summaries, developing decoding-aware training objectives, and introducing stronger prompt-structure cues (e.g., token delimiters) to better separate intent from context. Additionally, extending the framework to support neutral or mixed sentiment, or applying it to other domains like product or book reviews, offers promising directions for further research.

7. Acknowledgement

We would like to thank Professor Zoran Kostic for his guidance throughout the course. Additionally, we would like to thank the TAs, William and Ian, for their insights and consistent help throughout the semester to make this project possible.

8. References

[1] Github Link
[2] Final Presentation
[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, *Language Models are Unsupervised Multitask Learners*. OpenAI, 2019.
[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,"
[5] A. Vaswani *et al.*, "Attention Is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
[6] E. Hu *et al.*, "LoRA: Low-Rank Adaptation of Large Language Models," *arXiv preprint arXiv:2106.09685*, Jun. 2021.
[7] S. Q. Zhang *et al.*, "Parameter-Efficient Fine-Tuning of Pretrained Language Models: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*
[8] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*
[9] TextAttack, "textattack/bert-base-uncased-imdb," Hugging Face, 2020. [Online]. Available: https://huggingface.co/textattack/bert-base-uncased-imdb"
[10] S. Stilwell, "Explainable Prompt Learning for Movie Review Sentiment Analysis," Master's thesis, Univ. of Ottawa, Ottawa, ON, Canada, 2024. [Online]. Available: https://ruor.uottawa.ca/handle/10393/45657

9. Appendix

9.1 Individual Student Contributions in Fractions

	si2468	rpp2142	ap4617
Last Name	Iyengar	Patel	Pachaury
Fraction of (useful) total contribution	1/3	1/3	1/3

What I did 1	Research on Prior Work	Research on Prior Work	Research on Prior Work
What I did 2	BERT testing, Decoder inference pipeline	Dataset Curation	Creating datasets, preprocessing datasets for the inference pipeline,
What I did 3	Fine-tuning GPT	Fine-tuning GPT + LLaMA-2	Fine-tuning GPT + LLaMA-2