

From Premise to Punchline: A Fine-tuned Model and “Writers’ Room” Framework for Saturday Night Live Sketch Script Generation

Stanford CS224N Custom Project

Natalie Hampton
Department of Computer Science
Stanford University
nahamptn@stanford.edu

Hannah Yu
Department of Computer Science
Stanford University
hannahy9@stanford.edu

Abstract

Although recent LLM models have improved creative capabilities, creating genuine humor remains a challenge for models. In this project, we fine-tune a model and implement an agentic framework that generates high-quality Saturday Night Live (SNL) comedy sketch scripts. First, we supervised fine-tune Meta-Llama-3.1-8B-Instruct, a pretrained model, on pre-processed SNL scripts. Second, we implement a multi-agent framework mirroring the “writers’ room” process, involving agents for the Editor, Draft Reviser, and Audience roles. Our baseline model was Meta-Llama-3.1-8B-Instruct without finetuning. We use automatic evaluative metrics (Distinct-3, Self-BLEU, PPL) to compare the coherence and lexical diversity across baseline outputs, fine-tuned outputs, and writers’ room refined outputs. We also implement LLM-as-a-Judge to compare finetuned and refined outputs against the baseline, as well as finetuned outputs versus refined outputs. The judge evaluates in the dimensions of humor, coherence, and overall quality. We found that both the fine-tuned outputs and refined outputs improved over the baseline, with greater improvement in the refined outputs. These results suggest that while domain adaptation through fine-tuning improves humor generation, agentic revision frameworks based on human-collaborative processes offer an even more effective path toward higher-quality comedic content.

1 Key Information to include

- Mentor: Julie Kallini
- Late Days:

Team member	Late days remaining
<Natalie Hampton>	<6>
<Hannah Yu>	<0>
Total pooled	<6>
Team size	<2>
Late days applied for the report	<3>

2 Introduction

Humor is a uniquely human trait meant to evoke amusement through storytelling and departures from logic and the status quo. Humor often serves a social function by creating empathy and connection between audiences [1]. While large language models (LLMs) have already demonstrated strong capabilities in problem-solving and reasoning tasks, their performance in more creative realms, and particularly in comedic writing, remains a challenge [2, 3, 4]. A quintessential comedy hallmark —

Saturday Night Live — produces short comedy sketches that depend on unexpected premises, clear narrative setups, escalating comedic beats, and well-timed punchlines. While modern LLMs have been shown to be able to generate the structure of a script [5], the humor that is key to SNL sketches remains lacking.

In this paper, we explore two methods for improving LLM generation of SNL sketch comedy scripts. First, we fine-tune a Llama model on a dataset of SNL scripts to generate scripts from a given premise. Then, we implement a multi-agent framework meant to simulate a "writers' room" collaborative revision process to evaluate system-level architectural design choices in improving LLM script generations. Both methods are evaluated with a combination of automatic metrics and model-based evaluations. Our results show that both finetuning and writers' room framework improve the quality of generated sketches compared to a baseline model. Finetuning helps the model better capture SNL-style comedic structure, although sometimes struggles with consistently reproducing a desired script format. The writers' room framework consistently improves the comedy escalation and narrative flow of baseline scripts through structured critique and iterative revision.

3 Related Work

Though there is individual work into LLM humor generation and Collaborative Writing Frameworks to improve LLM script generation, there is limited work that investigates the specific intersection in sketch-comedy scripts. Our research attempts to bridge that gap.

3.1 LLM Humor Generation

Research shows that LLMs struggle with humor generation in multiple dimensions. First, when producing jokes, LLMs tend to reproduce puns that fit into learned patterns, indicating that the models are memorizing jokes instead of possessing genuine creative humor generation [4]. Furthermore, LLMs specifically struggle to generate and interpret humor in emotionally sensitive conversations. Research had shown that even when models can produce coherent humorous content, they still fall short in emotional realism and contextual appropriateness[2]. One paper found that LLMs specifically prioritize Novelty, while humans prioritize Empathy. This affects both the humor generated by the LLMs versus humans and the evaluation by LLMs-as-a-judge versus humans[3]. Emerging strategies have worked to bridge that gap. A reasoning framework called Creative Leap of Structured Thought (CLOST) encourages models to generate humor through structured reasoning chains, finding that using multi-step associative reasoning rather than just token prediction improves humor generation quality[6]. Furthermore, a modified version of Direct Preference Optimization (DPO) introduced a triple preference loss that compares positive humorous examples, negative examples, and structurally valid but less humorous examples, to help models maintain the correct pun structure while improving humor quality - based on the insight that models typically struggle to learn both tasks simultaneously. By handling multiple humor objectives, the modified DPO outperformed baselines[7].

3.2 Collaborative Writing Frameworks

Our writer's room approach fundamentally builds off the HoLLMwood framework [5], which is a process-oriented framework for automating screenwriting that organizes LLMs into multiple agents that take on different creative roles in the writers' room—specifically Writer, Editor, and Actors—each responsible for a distinct stage of the screenwriting pipeline with iterative feedback loops. Rather than proposing new architecture for models, the paper introduces a novel generation protocol that leverages existing LLMs more effectively. The paper found that HoLLMwood outputs were substantially better than the baselines in the dimensions of coherence, relevance, interestingness, and overall quality.

4 Approach

Our goal is to compare two methods for improving the quality of comedy sketch generations. First, we implemented a fine-tuning pipeline. Our goal is to learn a conditional generator:

$$p_{\theta}(y \mid x),$$

where x denotes the structured input (premise, setting, and cast) and y denotes the target formatted output (character list and multi-line script). We applied SFT to Llama-3.1-8B-Instruct [8] via QLoRA

[9]. We implemented a custom pipeline to preprocess the data so that it serializes each example into a single formatted string with explicit "Input" and "Output" sections. The formatting logic and data loading code were written by us and enforce consistent conditioning and output structure during training, ensuring that at inference time the model reliably generates the structured output section given the input section.

Since the model has 8B parameters and we face memory constraints, we used QLoRA for our fine-tuning, which combines 4-bit NF4 quantization via bitsandbytes with low-rank adaption (LoRA). In this setup, the base model weights are loaded in 4-bit format (whereas typically weight parameters are stored in 16-bit or 32-bit format) and remain frozen throughout training. LoRA constraints updates using low-rank decomposition, where

$$W = W_0 + aBA,$$

and where

$$B \in \mathbb{R}^{d \cdot r}, A \in \mathbb{R}^{r \cdot k}, r \ll \min(d, k).$$

A and B contain trainable parameters, and a is the constant scaling factor. We insert these trainable LoRA adapters into the attention and MLP projection layers, while the quantized base weights remain fixed. Thus, only the adapter parameters are tuned during training. We implemented this fine-tuning process using HuggingFace Transformers, PEFT for LoRA, bitsandbytes for quantization, and TRL's SFTTrainer (without any modification of these libraries). Our contribution consist of the dataset formatting pipeline, prompt template design, hyperparameter configuration, and the end-to-end fine-tuning set up.

Our second goal was to evaluate the effects of a multi-agent framework that simulates the iterative revision process seen in a real "writers' room." This draws upon the ideas introduced in Chen et al. [5], however unlike their framework, our system does not include a writer agent. Instead, we use the baseline generated scripts as the initial draft, and then pass it through a sequence of agents that assess and selectively revise the script. Our implemented framework consists of three agents: an Editor, an Audience Member, and a Draft Reviser.

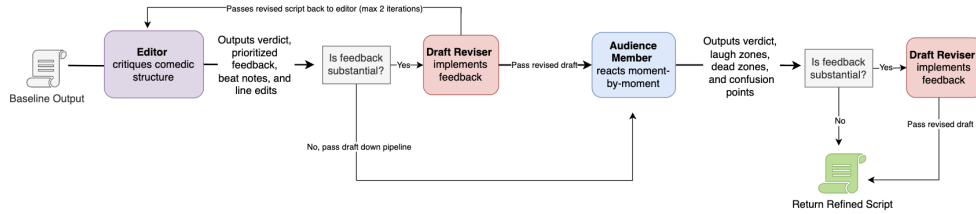


Figure 1: Writers' Room Workflow

- **Editor** analyzes the script for narrative clarity around the given premise, comedic escalation, and structural coherence. Its output contain (1) a summary of critiques, (2) list of specific issues such as unclear setups, weak punchlines, pacing problems, and (3) list of specific revision suggestions. The Editor can run for up to two rounds, but terminates early if suggested revisions are considered unsubstantial.
- **Audience** evaluates the script from a viewer's perspective, focusing on how jokes land, whether comedic beats escalate effectively, and if jokes on sensitive topics should be flagged. Its output contain (1) a map of reactions identifying if moments are funny, confusing, or weak, (2) a list of targeted suggestions to improve comedic escalation and structure, and (3) a list of any jokes about protected groups and their corresponding risk flag. This stage simulates the trial audience used during the SNL writing process where sketches are tested and jokes that do not receive strong reactions are revised or removed. The audience agent runs once and only passes on feedback to draft reviser if it is substantial.
- **Draft Reviser** is responsible for implementing substantial feedback from Editor and Audience. It receives the most recent script after the last stage of iteration and generates an updated version of the script that addresses all feedback notes.

As a baseline, we evaluated Llama-3.1-8B-Instruct in its pretrained form without any task-specific fine-tuning. We evaluate this baseline by prompting it with the same structured input format as our fine-tuned model via one-shot learning to ensure a fair comparison. This experimental set up allows us to compare the improvements to the baseline model between our two approaches of domain adaption via supervised fine-tuning and multi-agent workflows.

5 Experiments

5.1 Data

The dataset consists of input/output pairs derived from SNL sketch transcripts. The **input** is a JSON object with three fields: Premise (high-level description of the act), Setting (the location(s) and time period of the script), and Cast (the number of characters). The **output** is plain text beginning with a character list of names and roles, and the full cleaned script. Initially, we organized the data into the following highly-structured JSONs:

```
{
  "input": {
    "premise": "...",
    "setting": "...",
    "cast": "..."
  },
  "output": {
    "characters": [{
      "name": "...",
      "role": "...",
    }],
    "script": [...]
  }
}
```

However, we found that the increased structure reduced the quality of the outputs, as it prioritized complex formatting over coherence. This observation is consistent with recent research that shows LLMs, specifically Meta Llama models, struggle to generate structured outputs that align with required precision and constraints [10]. Although the research used older Llama models (Llama 2-13B and Llama 2-7B), it demonstrated the poorest performance among Llama models compared to GPT and Qwen Series models, "indicating that [Llama models'] advantage in dialogue optimization may not directly translate to exceptional performance in structured output tasks."

We then simplified the outputs to plain-text formatting:

```
CHARACTERS:
Character Name | Their role
Character Name | Their role

SCRIPT:
[Stage direction]
Character Name: Dialogue line
Character Name: Dialogue line
...
```

For analysis and consistency with our existing evaluation scripts, we processed and stored the outputs in structured JSONs, but the model was not exposed to these.

To construct this dataset, we first scraped a database of previous SNL scripts [11], collecting scripts from 2021, 2022, and 2023. We then manually labeled four examples—formatting the character list, cleaning the script, and writing the input metadata—to use as in-context learning. These examples were used to prompt "gemini-2.5-flash-lite" [12] to process the remaining scripts into a structured format. The final dataset contains 629 labeled sketches and uses a 90/10 split for training and testing (training on 569 and testing on 60).

5.2 Evaluation method

5.2.1 Automatic Evaluations

Though automatic evaluations cannot measure the quality of humor, which requires a subjective judge, they can serve as a proxy for the overall quality of the output. We used **Distinct-3** to measure the lexical variety in output - specifically the proportion of unique trigrams - where $\text{Distinct-3} = \frac{|\bigcup_{i=1}^{N-2} \{(w_i, w_{i+1}, w_{i+2})\}|}{\sum_{i=1}^{N-2} 1} = \frac{\text{unique 3-grams}}{\text{total 3-grams}}$. A higher score indicates greater variety. We used **Self-BLEU** to measure the diversity of outputs. The traditional BLEU calculation is

$$\text{BLEU} = BP \cdot \exp \left(\sum_{n=1}^N w_n \log p_n \right), \quad BP = \begin{cases} 1 & \text{if } c > r \\ e^{1-r/c} & \text{if } c \leq r \end{cases}$$

and is used to capture the similarity between reference text and model-generated output. Self-BLEU calculates the BLEU score of each output compared to the others, and averages the scores [13]. Lower Self-BLEU indicates there is greater diversity in the scripts. Finally, we used **Perplexity (PPL)** to measure the fluency and coherence of the output, where $PPL(W) = \exp \left(-\frac{1}{N} \sum_{i=1}^N \log P(w_i | w_1, \dots, w_{i-1}) \right)$ [14]. Lower values indicates higher probability of the generations, which can act as a proxy for naturalness.

5.2.2 LLM as a Judge

To assess the subjective quality of humor of our generated outputs, we used a model-based evaluation approach. Although prior work has shown that LLMs and humans do not fully align on humor detection [15], we decided to rely on this automatic evaluation due to scalability considerations and time constraints with human annotation. We implemented our LLM-as-a-judge framework using the Gemini 2.5 Flash-Lite [12, 3] API. We conducted three separate evaluations. First, we compared outputs generated by the fine-tuned model against the baseline model. Second, we compared outputs refined by the multi-agent framework (referred to as refined outputs) against the baseline outputs. Finally, we compared outputs generated by the fine-tuned model against the refined outputs.

For each comparison, the judge model was provided with both candidate scripts and instructed to evaluate and select the better script along three dimensions: *Humor*, *Coherence*, and *Overall Quality*. The *Humor* criteria measures the effectiveness of the comedic structure, including escalation of tension and the presence of distinct comedic character voices. *Coherence* evaluates narrative clarity, logical flow, and consistency. *Overall Quality* encompasses the holistic entertainment value and creativity of the sketch, overall reflecting which script would perform better as a piece of sketch comedy.

5.3 Experimental details

We had three experimental conditions, each with a different configuration.

1. **Baseline:** We directly loaded Meta Llama-3.1-8B-Instruct model from Hugging Face [8] without any additional training.
2. **Fine-tuned Model:** We used SFT with QLoRA [9] adapters on a dataset of structured SNL scripts on the baseline model. The base model was quantized to 4-bit precision using the BitsAndBytes implementation. We configured LoRA as follows: rank = 16, $\alpha = 32$, dropout = 0.05, target modules: attention and MLP layers. Training Hyperparameters: 4 epochs, learning rate of 1×10^{-4} , cosine scheduler, effect batch size = 8. Training was conducted using the TRL SFTTrainer [16]. The dataset was split into training and testing sets and formatted using the Llama model’s preferred chat template (system, user, assistant).
3. **Multi-agent Framework** We set up the writers’ room multiagent framework as described in the Approach section by creating three agents: an *editor*, a *draft reviser*, and an *audience member*. The editor and draft reviser iterate for at most two rounds when substantial revisions are proposed, followed by a single round of audience feedback and optional final revision. Draft revisions are generated with temperature = 0.35, top- p = 0.9, repetition penalty = 1.03, and a maximum of 1500 new tokens.

Using the same base model across all experimental conditions ensures that differences in performance arise from the training and refinement strategies rather than model architecture.

5.4 Results

Metric	Baseline	Fine-tuned	Writers' Room
Distinct-3	0.9981	0.9955	0.9683
Self-BLEU	0.0126	0.0092	0.0192
Perplexity	53.4716	39.4724	23.1800

Table 1: Automatic Evaluations: Comparison of baseline, fine-tuned, and writers' room outputs performance.

Metric	Baseline vs. Fine-tuned		Baseline vs. Writers' Room		Writers' Room vs. Fine-tuned	
	Baseline	Fine-tuned	Baseline	Writers' Room	Writers' Room	Fine-tuned
Humor	5	55	0	60	52	8
Coherence	10	50	0	60	52	8
Overall Quality	4	56	0	60	52	8

Table 2: LLM as a Judge: Pairwise comparisons across all model pairs.

As seen in Table 1, the baseline model has a slightly higher Distinct-3 score than the fine-tuned model (.0026) and the writers' room model (.0298), which indicates the baseline model had the highest lexical variety and the writers' room model had the smallest. The Self-BLEU score was lower by .0034 in the fine-tuned model than the baseline, which indicates there was greater diversity between the scripts, and higher by 0.0066 in the writers' room, which suggests that the refined scripts are more similar to each other in wording. This is expected however because iterative revision moves towards standardizing phrasing and narrative structure to improve clarity and cohesion. However, both these differences are small enough that they indicate an overall similar capability of lexical variety and diversity in outputs. The fine-tuned model's Perplexity score was 12.992 less than the baseline model's score, and the writers' room model's score was 30.2916 less. This indicates that both the fine-tuned and writers' room outputs were more fluent and natural-sounding than the baseline, with a greater improvement in the writers' room outputs.

The LLM-as-a-Judge evaluations in Table 2 show a significant improvement in the higher-level qualities of humor, coherence, and overall quality in both the fine-tuned and writers' room outputs. When comparing the baseline and fine-tuned outputs, the judge significantly preferred the fine-tuned model. Although for 10/60 of the scripts the judge thought the baseline output was more coherent than the fine-tuned, the judge preferred the overall quality of the baseline only for 4/60 examples. This indicates that the fine-tuned model may be improving humor at the expense of coherence. The judge preferred the writers' room over the baseline in all scripts in all three criteria, and preferred the writers' room over the fine-tuned in 52/60 scripts. This supports that both finetuning and writers' room workflows improve outputs, but the multi-agent framework centered around directed feedback and iterative revision is especially effective at improving comedic escalation and narrative flow, generating better comedy sketches than a baseline LLM.

Metric	Baseline	Post-Editor-1	Post-Editor-2	Post-Audience
Perplexity	49.49	28.67	24.45	22.45
Distinct-3	0.9977	0.9860	0.9685	0.9630
Self-BLEU	0.0138	0.0167	0.0178	0.0178

Table 3: Automatic Evaluations: Comparison of metrics across writers room pipeline stages.

By collecting scripts at each stage of the writer's room, we also can compare in Table 3 the value added by each round of iteration using automatic evaluations. Most significantly, the Perplexity decreased through each round of iteration, but the magnitude it decreased by reduced each round. This suggests that though additional rounds of iteration help to improve the coherence of the script, there are diminishing returns. This motivated our approach to make editor feedback optional and only have the draft reviser implement it if it was considered substantial.

6 Analysis

6.1 Fine-tuning Analysis

The fine-tuned outputs indicate the importance of the data. For example, as discussed in **5.1: Data**, training on highly-structured data reduced the quality of the outputs. It may have been overfitting to the structure of the JSON. For example, when outputting the list of characters, an excerpt from an early iterations of the fine-tuned model returned:

```
{...\"name\": \"Joe Biden\", \"role\": \"U.S. President\"}, {\"name\": \"Elon Musk\", \"role\": \"Businessman\"}, {\"name\": \"Joe Biden\", \"role\": \"U.S. President\"}, {\"name\": \"Elon Musk\", \"role\": \"Businessman\"}, {\"name\": \"Joe Biden\", \"role\": \"U.S. President\"}, {\"name\": \"Elon Musk\", \"role\": \"Businessman\"}, {\"name\": \"Joe Biden\", \"role\": \"U.S. President\"}, {\"name\": \"Elon Musk\", \"role\": \"Businessman\"}...
```

Though changing the format to plain text helped, the fine-tuned model still demonstrated a general optimization for humor at the expense of coherence, as indicated by the results of the LLM-as-a-judge. Several outputs devolved into runaway generation, for example the following dialogue line: Self-awareness allowed me to realize that I become dependent on other substances besides narcotics including caffeine, sugar, nicotine, alcohol, marijuana, mushrooms, LSD, ecstasy, ketamine, PCP, GHB, Rohypnol, OxyContin, Vicodin, Percocet, Adderall, Ritalin, Xanax, Valium, Ambien, Lunesta, Sonata, Restoril, Dalmane, Estazolam, Quaaludes, DMT, psilocibin, ayahuasca, ibogain, peyote, Salvia divinorum, Kratom, Kava kava, CBD oil.... The runaway generation typically occurred in places where it was logical to have some short list, but the error arose when the model did not correctly exit out and continue the line. In SNL skits, repetition is often used as a deliberate stylistic-choice to escalate absurdity. The model may have learned to escalate the local repetition patterns without learning when to stop.

The fine-tuned model also learned specific themes from the training data. For example, though only 4 of the 60 input premises for the testing data mention COVID-19, it is mentioned in 9 of the output scripts - likely due to the significant references to COVID-19 in the years 2021, 2022, and 2023 that we trained on data from. Though training on a wider range of years could reduce the overfitting to any particular theme or time period, it also could make the references less timely and relevant.

6.2 Writer's Room Framework Analysis

One of the ways in which the writers' room framework improves humor and coherence is by connecting fragmented narrative pieces to a central or recurring theme and also escalating underdeveloped comedic beats from the baseline output. We can clearly see this in the following examples about a "Beanie Babies" sketch, where a toy company employee writes increasingly disturbing backstories for children's toys. In the baseline output, the premise appears briefly in lines such as:

```
{... Peanut was born during an exorcism...}
```

While unexpected and absurd which can make the line humorous to the audience, the idea is then not strongly developed throughout the rest of the scene and the comedic beat dies there. By contrast, the refined version after the writers' room pipeline repeatedly escalates the same comedic theme. For example, the toy maker introduces his work by saying:

```
{... I'm crafting a masterpiece that will traumatize children worldwide....}
```

and later rationalizes it with:

```
{... What do kids know about existential dread and the meaninglessness of consumerist society?...}
```

The sketch culminates with an even more exaggerated toy description:

```
{... Patti Wocket’s favorite hobby is collecting the ashes of extinct  
species....}
```

This reinforcement and escalation of a central joke in the refined version, along with increasingly concerned reactions from coworkers, creates a more cohesive and intense comedic structure than the baseline. This makes sense given that the Editor and Audience agents are instructed to focus on comedic beat escalation and identify comedic dead zones, and provide suggested edits to the Draft Reviser agent.

However, we noticed that the refined outputs were limited in that the writers’ room agents tended to revise scripts using similar feedback patterns. This means that outputs sometimes fell back on similar common comedic devices, such as exaggerated props and chaotic physical comedy which was notable through the stage directions. This may reduce comedic diversity *across* sketches, which over time would make SNL skits more predictable and boring to audiences. Overall, it is clear that the Writers’ Room framework significantly improves coherence and comedic escalation by constantly reinforcing a central joke and strengthening narrative structure around comedic beats, even though the process can lead to some stylistic standardization across outputs.

6.3 Ethical Considerations

Our project raises a few ethical considerations, relating to intellectual property (IP) rights and the generation of harmful humor. The primary concern involves copyright and IP rights. *SNL* scripts are copyrighted works owned by NBCUniversal and the IP of these scripts are protected. The transcripts used to fine-tune our baseline model originate from these protected materials. While we used transcripts solely for research purposes, our process does reflect broader considerations of using copyrighted creative content as training for LLMs and the boundaries of fair use in ML development. A second ethical consideration relates to nature of human itself. Comedy frequently relies on engagement with social identities and stereotypes related to race, gender, class, and sexuality. Humans can rely on contextual judgment, audience awareness, and delivery tactics to navigate this thin boundary between satire and offensive content. LLMs lack this social context, which increases the risk that generated sketches may produce jokes that reinforce harmful stereotypes. Further research demonstrates that designing LLMs to optimize humor leads to a perpetuation and amplification of stereotypical and toxic outputs [17]. Specifically in the context of our writers’ room, the research indicates that since LLMs tend to rate stereotypical and toxic outputs as funnier, the multi-agent framework is at extra risk for generating harmful content. We attempted to mitigate this with our Audience agent by including sensitivity rules in the system prompt to flag sensitive content with appropriate risk levels throughout the writers room revision process. Other LLM tasks of generating comedic content should incorporate safeguards in system design and implement human review to ensure that generated content is not harmful to protected groups.

7 Conclusion

We were able to successfully demonstrate that fine-tuning models on SNL scripts and executing a writers’ room agentic workflow improve the quality of LLM-generated SNL skits, with the agentic framework offering the most effective path towards high-quality generation. Though our project focused on SNL scripts, the learnings can be applied to help improve the persistent problem of LLM humor broadly.

Our project separately evaluated the effect of fine-tuning a model and of utilizing a writers’ room framework to improve the quality of outputs. Future work could test using the multiagent framework on fine-tuned outputs, and even going a step further to fine-tune the editor, reviser, and audience agent models towards their specific roles. The primary challenge in accomplishing this is identifying and collecting what data to fine-tune on, since it requires additional metadata besides just the prompt and script. Future work could also refine the guardrails and investigate how to prevent the creation of harmful, toxic content by LLMs while still preserving the engagement with social themes that SNL sketches and humor requires.

Team contributions

Natalie: Wrote scripts for data pre-processing and automated evaluation for outputs. Refined parameters for fine-tuning through iterative evaluation.

Hannah: Implemented finetuning pipeline, writers’ room framework, and LLM-as-Judge evaluation.

References

- [1] Sumi Han Hwan Kim, Hyera Choi. The effect of sense of humor and empathy on the interpersonal adaption. *Personality and Individual Differences*, 2022.
- [2] Jessis Chin Kexin Quan, Pavithra Ramakrishnah. Can ai take a joke—or make one? a study of humor generation and recognition in llms. *Proceedings of the 2025 Conference on Creativity and Cognition*, 2025.
- [3] Tosho Hirasawa Mamoru Komachi Ritsu Sakabe, Hwichan Kim. Assessing the capabilities of llms in humor: A multi-dimensional analysis of oogiri generation and evaluation, 2025.
- [4] Kristian Kersting Sophie Jentzsch. Chatgpt is fun, but it is not fully! humor is still challenging large language models, 2023.
- [5] Jing Chen, Xinyu Zhu, Cheng Yang, Chufan Shi, Yadong Xi, Yuxiang Zhang, Junjie Wang, Jiashu Pu, Rongsheng Zhang, Yujiu Yang, and Tian Feng. Holmwood: Unleashing the creativity of large language models in screenwriting via role playing, 2024.
- [6] Dian Li Xiaohan Wang Gang Liu Xuguang Lan Hui Wang Han Wang, Yilin Zhao. Innovative thinking, infinite humor: Humor research of large language models through structured thought leaps, 2025.
- [7] Tu Hu1 Xinhao Chen Man Lan Li Cai Xinling Zhuang Xuan Lin Xin Lu Aiming Zhou Yang Chen, Chong Yang. Are u a joke master? pun generation via multi-stage curriculum learning towards a humor llm, 2025.
- [8] Meta AI. Llama-3.1-8B-Instruct. <https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct>, 2024. Accessed: March 2026.
- [9] Ari Holtzman Luke Zettlemoyer Tim Dettmers, Artidoro Pagnoni. Qlora: Efficient finetuning of quantized llms. In *NIPS ’23: Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023.
- [10] Kaili Wang Zhuoran Xiong Fobo Shi Jian Wang Bing Li Bo Hang Yu Liu, Duantengchuan Li. Are llms good at structured outputs? a benchmark for evaluating structured output capabilities in llms. *Information Processing and Management*, 2024.
- [11] Don Roy King. SNL transcripts: Saturday night live transcripts. <https://snltranscripts.jt.org>. Accessed: 2026.
- [12] Google Cloud. Gemini 2.5 flash-lite. <https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-flash-lite>, 2026. Accessed: March 2026.
- [13] Ehsan Montahaei, Danial Alihosseini, and Mahdieh Soleymani Baghshah. Jointly measuring diversity and quality in text generation models. In *Proceedings of the AAAI Workshop*, 2019.
- [14] Perplexity of fixed-length models. <https://huggingface.co/docs/transformers/perplexity>. Accessed: 2026.
- [15] Piotr Mirowski, Juliette Love, Kory Mathewson, and Shakir Mohamed. A robot walks into a bar: Can language models serve as creativity support tools for comedy? an evaluation of llms’ humour alignment with comedians. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT ’24, pages 1622–1636, New York, NY, USA, 2024. Association for Computing Machinery.
- [16] Younes Belkada. Sft trainer. https://huggingface.co/docs/trl/sft_trainer, 2025. Accessed: March 2026.

- [17] Ameet Deshpande Ashwin Kalyan Dinesh Manocha Atharvan Dogra, Soumya Suvra Ghosal. Engagement undermines safety: How stereotypes and toxicity shape humor in language models, 2025.