Humor Generation

Sina Eskandari
s343349

Seyed Amirreza Mousavi
s326447

Amirreza Rahimi
s326964

Mona Pouresmaeil
s329609

Marcello Vitaggio
s318904

Abstract

Humor generation presents significant challenges due to subjectivity and the limitations of automatic metrics. In this work, we address SemEval Task 1 Subtask A by evaluating three instruction-tuned models (Llama 3.1, Gemma 2, and Qwen 2.5) via a round-robin LLM judging framework. We investigate the impact of retrieval-augmented generation (RAG) and Direct Preference Optimization (DPO) on performance. Our results identify Llama 3.1 as the strongest baseline and demonstrate that DPO consistently improves humor quality across configurations. These findings confirm the efficacy of LLM-based judging as a practical training signal for optimizing subjective generation tasks.

1 Introduction

Humor is a fundamental aspect of human communication, enabling social bonding and creativity. Generating humor, however, remains challenging for Natural Language Processing systems because comedic success depends on subtle semantic incongruity, cultural knowledge, and highly subjective human preferences. Unlike tasks such as translation or summarization, humor lacks a clear notion of correctness, making both modeling and evaluation difficult.

Recent advances in large language models (LLMs) have improved fluency and coherence in text generation, and modern instruction-tuned models can produce jokes and wordplay. Nevertheless, humor generation still faces two key limitations: automatic metrics correlate poorly with perceived funniness, and human evaluation is expensive and hard to scale. The SemEval Humor Generation shared task addresses this challenge by evaluating systems through human pairwise preferences under explicit generation constraints, but this setting also removes access to labeled data or explicit reward signals.

In this work, we investigate whether small instruction-tuned LLMs can be further improved in this unsupervised setting by using LLMs themselves as scalable judges and training signals. We compare three open models using a round-robin self-judging framework, study the effect of retrieval-augmented generation, and apply Direct Preference Optimization (DPO) using synthetic preferences derived from LLM judgments.

Our contributions are:

- A scalable LLM-based evaluation framework for constrained humor generation.
- An empirical comparison of three small instruction-tuned LLMs and the impact of retrieval augmentation.
- An application of DPO with synthetic preferences for improving humor generation in a subjective task.

2 Related Work

2.1 Humor Generation and Evaluation

Computational humor has evolved from template-based systems to modern Transformer-based LLMs with improved semantic flexibility. However, evaluating humor quality remains challenging due to its subjectivity and the poor correlation of automatic metrics with human judgments. Recent work has employed LLMs as judges to provide scalable pairwise preference evaluations, offering a practical alternative to expensive human annotation while maintaining consistency.

Retrieval-augmented generation (RAG) has shown promise in grounding generation with external knowledge for tasks like question answering. Its application to humor generation, where external context could enhance topical relevance and coherence, remains underexplored.

2.2 Preference Learning & DPO

Language model alignment increasingly relies on preference learning through pairwise comparisons. Reinforcement Learning from Human Feedback (RLHF) trains a reward model on preferences, then optimizes using PPO, but this approach is complex and unstable.

Direct Preference Optimization (DPO) (Rafailov et al., 2024) simplifies this by directly optimizing the model to prefer better responses using a closed-form objective, eliminating the explicit reward model. Our work extends DPO to humor generation using synthetic preferences from LLM judges, investigating whether model-generated feedback can improve performance in subjective creative tasks.

3 Task Description

3.1 Task Definition

We address Subtask A (Text-based Humor Generation) of the SemEval shared task, where the goal is to generate a short joke j given a prompt p. The system must satisfy one of two constraints: **Word Inclusion**, which requires the joke to contain two predefined rare words, or **News Headline**, where the joke serves as a humorous reinterpretation of a given news title. Unlike standard generation tasks, this problem is inherently subjective and lacks well-defined automatic evaluation metrics.

3.2 Evaluation Setting

The official evaluation relies on pairwise human preference judgments rather than automated metrics like BLEU or ROUGE, which correlate poorly with humor quality. Crucially, the provided dataset lacks gold labels or humor scores, making the task inherently unsupervised. This absence of ground truth prevents standard supervised training, compelling approaches to rely on weak supervision or indirect signals to optimize for perceived funniness.

4 Methodology

4.1 Low-Rank Adaptation (LoRA)

To address the high computational cost of full-parameter fine-tuning, we employ Low-Rank Adaptation (LoRA). This method freezes the pre-trained weights $W_0 \in \mathbb{R}^{d \times k}$ and optimizes trainable rank-decomposition matrices $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$ (where $r \ll \min(d, k)$), based on the hypothesis that adaptation updates have a low intrinsic rank. The weight update is approximated as $\Delta W = BA$, resulting in the modified forward pass:

$$h = W_0 x + BAx \tag{1}$$

We initialize A with a Gaussian distribution and B with zeros to ensure the starting state matches the pre-trained model ($\Delta W = 0$). We then optimize only A and B to minimize the objective function:

$$\min_{A,B} \sum_{(x,y) \in \mathcal{D}} \mathcal{L}(f(x; W_0 + BA), y) \tag{2}$$

A scaling factor $\frac{\alpha}{r}$ is applied to the update to stabilize learning and facilitate seamless merging of adapters during inference.

4.2 RAG-based Humor Generation

In this work, we employ a retrieval-augmented generation (RAG) approach to enhance humor generation by incorporating external context related to the input lines. The external context is retrieved from a pre-embedded Wikipedia corpus available on Hugging Face (not-lain, 2023), using a subset of 25,000 documents. For each input headline or word-pair, the top-4 most relevant documents are retrieved and included in the generation context. Query embeddings are computed using a sentence embedding model (Mixedbread AI, 2024).

The retrieved content is incorporated into the model input to support more coherent and contextually relevant joke generation. To further guide the generation process, retrieval is combined with a one-shot prompting strategy, where the model is provided with a single example of the desired output style. Additionally, a lower sampling temperature of 0.7 is used to reduce excessive randomness, encouraging the model to rely more strongly on the retrieved context while preserving creative variation.

4.3 Direct Preference Optimization

4.3.1 Motivation

Preference learning is commonly implemented using Reinforcement Learning from Human Feedback (RLHF), which consists of first training a reward model from human preference data and then optimizing a policy using reinforcement learning under a KL-divergence constraint. While effective, this pipeline is complex and often unstable, requiring the training of multiple models and sampling-based policy optimization.

Direct Preference Optimization (DPO) provides a simpler alternative by showing that the standard RLHF objective can be optimized without explicit reward modeling or reinforcement learning. Instead, DPO directly optimizes the language model policy using a supervised objective derived from the same preference assumptions as RLHF, resulting in a stable and computationally lightweight training procedure.

4.3.2 Preference Dataset

DPO operates on a dataset of pairwise preferences of the form:

$$D = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{i=1}^{N} \tag{3}$$

where x is a prompt, y_w is the preferred (winner) completion, and y_l is the rejected (loser) completion. In our setting, this dataset is constructed synthetically using LLM-based judges, but the optimization framework remains identical to the original DPO formulation.

4.3.3 DPO Objective

DPO is derived from the same constrained RL objective used in RLHF:

$$\begin{aligned}\max_{\pi_\theta} \quad & \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(y|x)}\big[r_\phi(x,y)\big] \\ & -\ \beta\,\mathbb{D}_{\mathrm{KL}}\big[\pi_\theta(y \mid x) \,\|\, \pi_{\mathrm{ref}}(y \mid x)\big]\end{aligned} \tag{4}$$

where π_θ is the policy being optimized, π_{ref} is a fixed reference policy, $r_\phi(x,y)$ is a reward function, and β controls deviation from the reference model.

(Rafailov et al., 2024) shows that the optimal solution to this objective can be obtained directly by optimizing the policy with a binary classification loss over preference pairs, without introducing an explicit reward model or reinforcement learning loop.

Explicitly, the DPO loss encourages the model to assign higher probability to preferred completions than to rejected ones, relative to the reference model:

$$\begin{aligned}\mathcal{L}_{\mathrm{DPO}}(\pi_\theta;\pi_{\mathrm{ref}}) = &-\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\\ & \log\sigma\Big(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} \\ & -\beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)}\Big)\Big]\end{aligned} \tag{5}$$

where σ is the logistic function.

4.3.4 Application to Humor Generation

In our work, we apply DPO using synthetic preferences produced by LLM-based judges rather than human annotators. While the original formulation assumes human preferences, the optimization framework remains unchanged. The model is therefore trained to align with the preference distribution induced by the judges, allowing us to apply preference learning in a fully unsupervised manner.

This enables us to investigate whether DPO can improve performance in a highly subjective creative task, such as humor generation, using only model-generated feedback.

5 Experiments

5.1 LoRA Fine-Tuning Setup

To evaluate the effectiveness of Supervised Fine-Tuning (SFT) for humor generation, we conducted two distinct experiments using Low-Rank Adaptation (LoRA) on the Llama-3.1-8B-Instruct base model. LoRA allows us to efficiently update a small subset of parameters while freezing the pretrained weights, making it ideal for our computational constraints.

5.1.1 Experiment 1: Synthetic Data Fine-Tuning

The first experimental setup focused on enhancing the model's reasoning capabilities using a custom, synthetically enriched dataset.

Data Construction We derived this dataset from the Short Jokes Dataset. The raw corpus first underwent deduplication and length-based filtering to remove outliers. To prevent the model from overfitting to specific joke structures, we applied stratified sampling based on length bins. We then utilized an LLM-based generation pipeline to synthetically enrich each entry, transforming raw jokes into structured triplets containing a clear instruction, a generated reasoning trace explaining the humor, and a refined joke text. This process resulted in a robust dataset of approximately 3,500 samples.

Training Configuration We fine-tuned the Llama-3.1-8B-Instruct base model using Low-Rank Adaptation (LoRA) with 4-bit quantization. The LoRA configuration used a rank (r) of 16 and a scaling factor (α) of 16. To ensure the model internalized the reasoning structure without catastrophic forgetting, we trained for **2 epochs**

with an effective batch size of 8.

5.1.2 Experiment 2: Best-of-N Distillation

The second setup explored a "distillation" approach, leveraging the best outputs from our initial multi-model tournament.

Data Construction We constructed a "Best Jokes" dataset by aggregating the top-performing jokes generated by three base models (Llama, Qwen, and Gemma). The source pool consisted of 1,200 distinct joke scenarios. For each scenario, the winner of a pairwise comparison (judged by an independent model) was selected. This yielded a high-quality dataset of 1,200 samples representing the peak capabilities of the base models.

Training Configuration Using the same Llama-3.1-8B-Instruct base, we applied a distinct training strategy optimized for this smaller, high-quality dataset. We trained for **2 epochs** using a learning rate of 2×10^{-5} with a linear scheduler. The effective batch size was increased to 16 (per-device batch size of 4 with gradient accumulation of 4). Optimization was performed using Paged AdamW (8-bit) to maximize memory efficiency.

5.2 Setup: Direct Preference Optimization (DPO)

For the preference learning phase, we selected Llama-3.1-8B-Instruct as the base model due to its superior performance in our initial generation tasks.

Training Configuration We employed parameter-efficient DPO, updating only the LoRA adapters (Rank 16, α 16) while keeping the base model frozen in 4-bit precision. Training was conducted with a conservative learning rate of 5×10^{-6} for a single epoch, utilizing 50 warmup steps to stabilize the loss.

Hyperparameters The DPO loss incorporated a KL-regularization parameter $\beta = 0.05$. This value was chosen to encourage the model to learn from preferences while preventing excessive deviation from the original policy. The effective batch size was set to 8, and training was carried out in FP16 precision.

5.3 Setup: Evaluation Framework

To ensure a consistent and reproducible assessment, we established a standardized evaluation pipeline.

Judge Model We utilized a pairwise evaluation framework consistent with the official SemEval protocol. An independent Gemma model served as the judge for all comparisons. The judge was kept entirely separate from the training process to maintain objectivity.

Protocol For each evaluation, two systems generated a joke for the same prompt. The judge model was presented with the prompt and both anonymized candidates, then tasked with selecting the more humorous and coherent option. Performance is reported as the "win rate" of the fine-tuned model against two strong baselines: the original pre-trained Llama-3.1-8B-Instruct and the best-performing RAG-based system (Qwen).

5.4 DPO Fine-Tuning Setup

We apply Direct Preference Optimization (DPO) to fine-tune the strongest base model identified in the initial tournament, namely Llama-3.1-8B-Instruct. While our earlier experiments compare three models, only Llama is used for DPO fine-tuning due to its superior performance as the joke generator.

To enable efficient training, we employ parameter-efficient fine-tuning using Low-Rank Adaptation (LoRA). The base model is loaded using 4-bit quantization, and only LoRA parameters are updated during training.

5.4.1 Model and Training Configuration

We fine-tune Llama-3.1-8B-Instruct using a sequence length of 2048 tokens and 4-bit quantization, applying Rank-Stabilized LoRA. The configuration uses a rank of 16 with a scaling factor (α) of 16. During training, only the LoRA adapter parameters are updated, while all base model parameters remain frozen.

5.4.2 DPO Settings

DPO training is conducted with a per-device batch size of 4 and gradient accumulation over 2 steps (effective batch size of 8). We use a learning rate of 5×10^{-6} and train for a single epoch with 50 warmup steps. Optimization is performed using AdamW with 8-bit parameters and a weight decay of 0.01 in FP16 precision.

The DPO loss incorporates a KL-regularization parameter $\beta = 0.05$, encouraging effective learning from preferences while preventing excessive deviation from the base Llama model.

5.4.3 Evaluation Settings

We evaluate DPO by comparing the fine-tuned model against two strong baselines: the original Llama-3.1-8B-Instruct and the best-performing RAG-based system (Qwen). We follow the same pairwise evaluation framework used throughout the project. Given a prompt, two candidate jokes are generated by the systems under comparison. A Gemma model acts as an independent judge to select the preferred joke, ensuring the judge is blind to the source of the generation.

5.4.4 DPO Fine-Tuning Setup

We apply DPO to fine-tune the strongest base model identified in the initial tournament, namely Llama-3.1-8B-Instruct. While our earlier experiments compare three models (Llama, Qwen, and Gemma), only Llama is used for DPO fine-tuning due to its superior performance as the joke generator.

To enable efficient training, we employ parameter-efficient fine-tuning using Low-Rank Adaptation (LoRA). The base model is loaded using 4-bit quantization, and only LoRA parameters are updated during training, allowing fine-tuning on limited computational resources.

5.4.5 Model and Training Configuration

We fine-tune Llama-3.1-8B-Instruct using a sequence length of 2048 tokens and 4-bit quantization, applying Rank-Stabilized LoRA as the fine-tuning method. The LoRA configuration uses a rank of 16 with a scaling factor (α) of 16. During training, only the LoRA adapter parameters are updated, while all base model parameters remain frozen.

5.4.6 DPO Settings

DPO training is conducted with a per-device batch size of 4 and gradient accumulation over 2 steps, resulting in an effective batch size of 8. We use a learning rate of 5×10^{-6} and train for a single epoch, with 50 warmup steps. Optimization is performed using AdamW with 8-bit parameters and a weight decay of 0.01, while training is carried out in FP16 precision.

The DPO loss incorporates a KL-regularization parameter β , which controls the strength of the constraint that keeps the fine-tuned policy close to the reference model; we set $\beta = 0.05$, encouraging effective learning from preferences while preventing excessive deviation from the base Llama model.

5.4.7 Evaluation Settings

We evaluate DPO by comparing a DPO-finetuned Llama model against two strong baselines: the original pretrained Llama-3.1-8B-Instruct model and the best-performing RAG-based system from our earlier experiments, namely Qwen. For each comparison, both systems generate a joke for the same prompt, and a third language model acts as a judge to select the more humorous output.

We follow the same pairwise evaluation framework used throughout the project. Given a prompt, two candidate jokes are generated by the systems under comparison. A Gemma model is used as an independent judge to select the preferred joke. The judge is not involved in training either system.

Each comparison is conducted over a fixed set of prompts, and final performance is reported as the number of wins for each system.

This protocol is consistent with the official SemEval evaluation setting, which is based on human pairwise preference judgments.

6 Results

6.1 Baseline Model Evaluation

The quantitative results of the triangular tournament, summarized in Table 1, reveal a clear hierarchy in humor generation performance. **Llama 3.1 8B** emerged as the dominant generator, securing substantial victory margins when evaluated by both Gemma 2 (674 vs 524) and Qwen 2.5 (726 vs 474). In contrast, the competition between Gemma 2 and Qwen 2.5 was highly contested; adjudicated by Llama 3.1, Gemma achieved a narrow victory (596 vs 589) with 15 ties, indicating comparable capabilities. Overall, the judges exhibited high decisiveness with minimal ties across all rounds, establishing a final ranking of Llama 3.1 first, followed by Gemma 2, and Qwen 2.5 in third.

Judge Model	Contestant A	Score	Contestant B	Score	Ties
Llama 3.1	Gemma 2	**596**	Qwen 2.5	589	15
Gemma 2	Llama 3.1	**674**	Qwen 2.5	524	2
Qwen 2.5	Llama 3.1	**726**	Gemma 2	474	0

Table 1: Tournament Results: Pairwise Comparisons by Judge

6.2 LoRA Fine-Tuning: Experiment 1

In this phase, we evaluated Llama 3.1 8B fine-tuned on the "Best Jokes" subset of our Short Jokes Improved Dataset. Table 2 reveals a drastic performance degradation, with the base model achieving near-total dominance (930 vs. 270 under Qwen 2.5; 960 vs. 240 under Gemma 2). These extreme margins indicate that fine-tuning on this specific subset likely induced severe overfitting or catastrophic forgetting of the model's instruction-following capabilities.

Table 2: Comparison: Llama 3.1 Base vs. Llama FT (Best Jokes Subset)

Judge Model	**Model A**	**Score**	**Model B**	**Score**	**Ties**
Qwen 2.5	Llama Base	**930**	Llama FT (Best)	270	0
Gemma 2	Llama Base	**960**	Llama FT (Best)	240	0

6.3 LoRA Fine-Tuning: Experiment 2

We subsequently shifted our strategy to fine-tune on a "distilled" dataset composed of the best jokes generated by the base models during the baseline tournament. As shown in Table 3, the base model retained its superiority (690 vs. 498 under Qwen; 816 vs. 384 under Gemma). However, the significantly reduced defeat margin compared to Experiment 1 suggests that training on high-quality model outputs was more effective than the previous synthetic approach, even if still insufficient to surpass the strong priors of the base model.

Table 3: Comparison: Llama 3.1 Base vs. Llama FT (SemEval Best)

Judge Model	**Model A**	**Score**	**Model B**	**Score**	**Ties**
Qwen 2.5	Llama Base	**690**	Llama FT	498	12
Gemma 2	Llama Base	**816**	Llama FT	384	0

6.4 RAG-Enhanced Humor Generation

To evaluate the effectiveness of retrieval-augmented generation, we conducted the same triangular tournament evaluation using the Wikipedia-augmented models. The results reveal a significant shift in the performance hierarchy compared to the baseline models.

Judge Model	**Contestant A**	**Score**	**Contestant B**	**Score**	**Ties**
Qwen 2.5	Gemma 2	**670**	Llama 3.1	530	0
Llama 3.1	Gemma 2	249	Qwen 2.5	**951**	0
Gemma 2	Llama 3.1	293	Qwen 2.5	**906**	1

Table 4: RAG Tournament Results: Pairwise Comparisons by Judge

In stark contrast to baseline results where Llama dominated, RAG augmentation shifts superiority to Qwen 2.5, achieving decisive victories of 951 and 906 wins. Gemma 2 performs moderately (670 wins), while Llama's performance declines substantially. These findings suggest that retrieved context differentially benefits models, with Qwen exhibiting superior contextual exploitation for humor generation.

6.5 Impact of DPO Fine-Tuning

Table 5 summarizes the results of our DPO evaluation experiments. We compare the DPO-finetuned Llama model against two baselines: the original pretrained Llama model and the best-performing RAG-based system (Qwen+RAG). In both cases, a Gemma model is used as the judge.

Comparison	DPO Wins	Other Method	Ties	DPO Win Rate
DPO vs Base Llama	799	401	0	66.6%
DPO vs Qwen+RAG	827	369	4	68.9%

Table 5: Comparison of DPO performance against other methods

7 Conclusion

We investigated humor generation in a fully unsupervised setting, where no gold labels or automatic metrics are available. We introduced a scalable LLM-as-a-judge framework to both evaluate systems and construct synthetic preference data for optimization.

Our experiments highlight three main findings. First, among the evaluated instruction-tuned models, Llama 3.1 is the strongest baseline for constrained humor generation. Second, retrieval-augmented generation substantially changes the performance hierarchy, with Qwen 2.5 benefiting the most from external context. Third, Direct Preference Optimization with synthetic LLM judgments consistently improves humor quality, outperforming both the base model and the best RAG-based system.

These results confirm that LLM-based judging can serve as an effective training signal for subjective creative tasks. As future work, we plan to further analyze the role of retrieval by performing controlled ablations on the amount and structure of retrieved context, in order to better understand when and how external knowledge benefits humor generation.

References

Mixedbread AI. 2024. mxbai-embed-large-v1. Sentence embedding model.

not-lain. 2023. Wikipedia (embedded version). Pre-embedded Wikipedia corpus on Hugging Face.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.