Team TüLK at SemEval-2026 Task 1: Humor Generation with Qwen and Group Relative Policy Optimization

Konrad Brüggemann and Luting Hou
University of Tübingen
Department of General and Computational Linguistics
{ konrad-rudolf.brueggemann, luting.hou } @student.uni-tuebingen.de

Abstract

This paper addresses the challenge of computational humor generation proposed in SemEval-2026 Task 1: Humor Generation. Our approach leverages Group Relative Policy Optimization, with Qwen-2.5 serving as the policy and a custom joke rating model providing a powerful reward signal. We demonstrate that this framework is an effective and computationally efficient approach, reliably producing genuinely funny content that adheres to task constraints.

1 Introduction

The SemEval-2026 Task 1 requires participants to develop systems that generate humorous content under various constraints. Specifically, it contains two subtasks, and Subtask 1 focuses on text-based humor generation, whereas Subtask 2 focuses on multimodal humor generation (humorous captions for GIF images). Subtask 1 supports English, Chinese (Mandarin) and Spanish text, and the jokes have to be produced under the constraint that either a given word pair must appear in the joke or the joke must relate to a given headline. Evaluation is based on human preference derived from a crowd-sourced Elo leaderboard, where annotators select the funnier between two jokes.

We participate in Subtask 1, and our goal is to investigate whether reinforcement learning can improve the ability of large language models (LLMs) to generate humorous text. Consequentially, a key challenge is obtaining a robust reward signal that reflects the funniness of a joke. To address this, we train a humor classification model that produces bounded ratings for jokes, which are then incorporated into the reward function. Our results show that: 1) it is possible to train a neural network to predict how funny a joke is, and 2) the score generated by the model represents a useful reward signal in a reinforcement learning setup.

Using this framework, our system consistently generates humorous text from a given word pair or headline, while remaining computationally efficient enough to be reproduced on a single academic-grade GPU.

2 System Overview

We use Group Relative Policy Optimization (Shao et al., 2024) as the training algorithm in the reinforcement learning training. Our policy is an LLM of the Qwen-2.5 family (Yang et al., 2025). Given a prompt such as "Generate a funny joke that contains the words 'shoes' and 'microwave'." the model's response is then evaluated by the reward model, which returns a scalar reward that is to be maximized during training. In practice, the reward model aggregates a scalar funniness score from a trained humor-evaluation classification model (the Funniness Classifier) and several reward or penalty signals (the Heuristic Reward Signals).

2.1 Group Relative Policy Optimization

Group Relative Policy Optimization (GRPO) is a reinforcement learning algorithm proposed by DeepSeek (Shao et al., 2024) that efficiently trains LLMs by generating multiple responses and comparing them to determine the best one, removing the need for a separate value function. It uses group-based advantage estimation where the average reward of a group of answers serves as a baseline, allowing it to reward better-than-average responses while remaining computationally efficient. In our system, we use the GRPO implementation of the Transformers Reinforcement Learning (trl) package by Hugging Face, which is called GRPOTrainer. [1]

Practically, we prompt the policy model to generate a joke using a given word pair or headline. The completions are then evaluated by the reward model, which consists of a **funniness classifier**,

[1] https://huggingface.co/docs/trl/main/grpo_trainer

as well as **heuristic reward signals** to guide the formatting, diversity, and constraint satisfaction of the generated jokes, via small additional rewards or punishments. The following sections describe each component in detail.

2.1.1 Model Selection

Qwen2.5 is a series of open-source large language models by Alibaba Cloud, that were pre-trained on an extensive corpus of 18 trillion tokens (Yang et al., 2025). We select this architecture for its state-of-the-art efficiency, multilingual capabilities (supporting the task's three languages; Chinese, English, and Spanish), and smooth integration with other frameworks that were used in this work, including the Hugging Face trl library.[2]

During the development phase, we use **Qwen2.5-3B-Instruct**, which fits on academic grade GPUs such as the NVIDIA L40S or A100.[3] Qwen-2.5-3B-Instruct is a 3.09B parameter, instruction-tuned causal language model, which uses a transformer architecture with 36 layers and Grouped-Query Attention (GQA) (Ainslie et al., 2023). The attention component leverages 16 Query heads and 2 Key-Value heads to support a context length of 32,768 tokens.

2.1.2 Training Data

To train the Qwen-2.5-based model, we use 2000 news headlines from pnadel/nyt_headlines dataset [4], and randomly generate 2000 word pairs. Details of these data are provided in the appendix A.

2.2 Funniness Classifier

We are inspired by prior work (Goes et al. (2022)) showing that prompting LLMs to evaluate humor from multiple perspectives (e.g., preferences for self-defeating humor or aggressive humor) and aggregating these assessments approximate joke ratings by human evaluators. Adopting a similar strategy, we use scalar funniness scores derived from LLM judgments. We then train a classification model with these LLM ratings and use it as a reward signal. This approach not only improves efficiency but also increases determinism and reproducibility, as the learned model provides stable and consistent reward values.

2.2.1 Model Design

We train a Hierarchical Classifier architecture based on the XLM-RoBERTa-large encoder (Conneau et al., 2020), designed to predict a funniness score $s \in \{0, \ldots, 10\}$. The model is structured hierarchically, sharing the features extracted from the [CLS] token embedding, which are then passed through a non-linear projection layer. The prediction is decomposed into two heads: a Binary Head (P_B) that predicts whether the joke is "Not Funny" ($s = 0$) or "Funny" ($s \in \{1, \ldots, 10\}$), and a Child Head (P_C) that predicts the specific funniness level (1-10) for the Funny cases.

The model is trained using a composite loss function that minimizes both classification errors (Cross-Entropy) and the magnitude of scoring errors (Mean Squared Error), using the true label as the regression target. This dual-objective loss is dynamically weighted by two learnable parameters that are both initialized at 0.5. The goal of this approach is to ensure both high classification accuracy and a low Mean Absolute Error (MAE).

Though our approach differs in architecture, we note that (Simone and Cruz, 2020) have previously successfully trained a Convolutional Neural Network to predict the funniness of a joke on a bounded scale, and their work motivated us to pursue this path.

Component	Description
Transformer	`xlm-roberta-large`
Dropout	0.5
Projection	$h \to h/2$
Binary head	funny vs. non-funny
Child head	funniness levels 1–10
Output	11-class distribution (0–10)
Final score	Argmax

Table 1: Overview of the hierarchical joke funniness model architecture.

2.2.2 Training Data

Datasets of jokes that are labeled with some bounded funniness score, are not readily available. However, collections of unlabeled jokes are easier to find. Therefore, we first collect jokes from a variety of sources, and then use LLMs to generate scores for them.

The English data consists of 550,000 jokes from the r/Jokes Dataset (Weller and Seppi, 2020), and approximately 32,000 human-picked humorous

[2] https://huggingface.co/docs/trl/index

[3] Specifically, we use the latest checkpoint available at https://huggingface.co/Qwen/Qwen2.5-3B-Instruct at the time of writing.

[4] https://huggingface.co/datasets/pnadel/nyt_headlines

texts provided by (Tang et al., 2024), (Weller and Seppi, 2019), and (Misra and Arora, 2023). Additionally, we synthetically generate 10,000 jokes for underrepresented scores (LLMs tend to give either a very high or a very low score, causing a gap in the 4-6 range). A detailed data breakdown is provided in the appendix C.

To label the jokes, we use Llama-3 (Grattafiori et al., 2024) and GPT-OSS (Agarwal et al., 2025) for English and Spanish jokes, and an uncensored Qwen-2.5 model and DeepSeek-3.1 for Chinese.[5] The prompt we use is included in the appendix B, and the districution of the LLM-generated ratings is provided in the appendix D.

2.3 Heuristic Reward Signals

As we briefly touched on, the reward function R is a composite signal designed to optimize for funniness, constraint satisfaction, and stylistic diversity. Hence, in addition to the funniness reward provided by the classifier outlined in 2.2, we implement several straight forward heuristics.

1) The **formatting reward** ensures that the output is a valid, single joke by penalizing commentary or conversational artifacts, specific prefixes and conversational markers (e.g., "How about..." or "This joke touches on..."), and excessive length via line breaks. This is augmented by the length penalty, which discourages what we call "joke stacking" by severely penalizing outputs outside the 5-24 word range and smoothly penalizing those exceeding an optimal 16-word length.[6]

2) Because, over time, the policy tends to subscribe to only a few types of joke structures that in general get good ratings from the funniness classifier (particularly "Why did X? Because Y!"), we add a **structure diversity reward** that dynamically incentivizes the generation of underrepresented joke structures (extracted using simple regex patterns) based on the frequency within a sliding window of recent completions.

3) To enforce the Subtask's constraints, we implement a **word-pair adherence reward**, which provides a positive signal when the given word pair is successfully integrated, while the **headline adherence reward** encourages the model to produce jokes that are semantically aligned with a given headline.

4) Finally, a **coherence penalty** discourages the use of rare or technical terms. As a consequence, the final scalar reward R is computed as a weighted sum of the five signals, $R = \sum r_i$, where w_i are predefined reward weights.

3 Experimental Setup

3.1 Funniness Classifier

The XLM-RoBERTa joke rating model is fine-tuned for 200 epochs using the standard Hugging Face Trainer framework. To optimize training efficiency, we activate mixed-precision (FP16) and a warmup ratio of 0.1, training with a batch size of 32 and a learning rate of 2×10^{-5}, while monitoring performance on the validation set after every epoch. Particularly, we monitor both accuracy and mean absolute error (MAE). We argue that, since it is an **ordinal** classification task, both of these metrics are highly relevant. for example, predicting 1 instead of 8 has the same negative effect on accuracy as predicting 7 instead of 8, though the latter is almost correct and hence preferable, which MAE takes into account.

3.2 Joke Generation Model

We train for 2 epochs with a low learning rate of 1×10^{-6}. The process leverages vLLM (Kwon et al., 2023) for efficient generation with a maximum completion length of 64 tokens and a sampling temperature of 0.5. Further, we utilize a consistent batch size of 16 for training and evaluation. At each step, we sample 8 completions (corresponding to G in the original GRPO paper (Shao et al., 2024)).

4 Results

We report evaluation results for both the funiness classifier as well as the actual trained joke generation model, though the latter is more relevant for the actual task.

4.1 Funniness Classifier

We compare different fine-tuned and frozen encoder models on a held-out test set. The results, as shown in Table 2, justify our decision to use the fine-tuned XLM-RoBERTa model in the final GRPO training loop. The experimental setup is the same for all four approaches (outlined in Section 3.1).

[5] We use the model Qwen2.5-7B-Instruct-Uncensored from `https://huggingface.co/Orion-zhen/Qwen2.5-7B-Instruct-Uncensored`.

[6] During the development phase, we detected a behavior of the policy that can be considered "reward hacking," where several jokes are stacked behind each other in order to boost the reward.

Classifier	Accuracy	MAE
XLM-R-large	0.09	5.58
XLM-R-large (ft)	**0.43**	**1.46**
mDeBERTa-large	0.09	5.58
mDeBERTa-large (ft)	0.39	1.68

Table 2: Evaluation results for different text classifiers, with (ft) indicating that the transformer layers were fine-tuned too as opposed to just using the features.

4.2 Joke Generation

Evaluating the trained policy is not straightforward. Therefore, we consider the training to be successful if a) the cumulative reward generally goes up over the duration of training until it converges and b) the jokes generated by the policy become funnier in our subjective opinions. We consider criterion a) to be satisfied, as shown by the plot provided in figure 1. Overall, we also consider criterion b) to be justified, with the caveat that not all generations are funny. However, compared to the policy pre-GRPO, and also compared to much larger models such as ChatGPT, we do find our model to be 1) consistently more funny, 2) better at actually satisfying the task constraints, and 3) more creative. These findings are, of course, subject to our personal preferences (and biases) and readers may make their own conclusions, based on sample outputs provided in the appendix **??**.

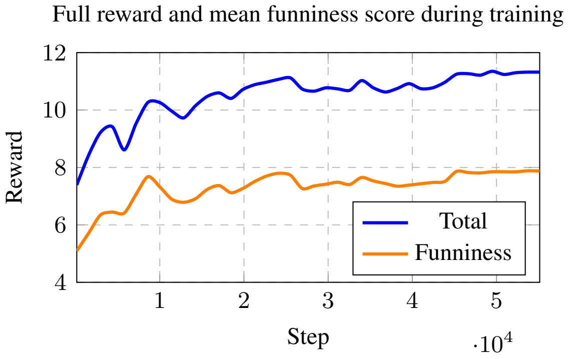

Figure 1: This plot shows the full training reward (blue) and the reward provided by the funniness classifier (orange) over the training steps (15-step rolling mean).

5 Interpretation

The increase and subsequent convergence of the cumulative reward in the GRPO training plot (1) provides the primary evidence that the reinforcement learning policy successfully optimized toward the composite reward function, which in turn suggests that the model learned to balance the (possibly) conflicting objectives of maximizing the funniness score (from the classifier) and satisfying the specific task constraints (word-pair adherence, formatting, and length). The relatively low MAE ≈ 1.4 of the funniness classifier confirms that the reward function can reliably differentiate between genuinely funny and non-humorous content. Qualitatively, the generated jokes demonstrate improved constraint adherence and structural diversity compared to Qwen-2.5 out-of-the-box. In our opinion, this confirms the efficacy of the system.

Limitations

The primary limitation is that this approach requires a synthetic funniness dataset, which may introduce label bias derived from the judging LLMs. While efficient, this may not perfectly align with the human preference metric (Crowdsourced Elo) used in the final SemEval evaluation. Furthermore, the many hand-tuned heuristic rewards (formatting, length, diversity) introduce hyperparameter complexity, which may affect how well the model generalizes outside of the specific constraints of the task.

References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, and 106 others. 2025. gpt-oss-120b gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *Preprint*, arXiv:2305.13245.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Fabricio Goes, Zisen Zhou, Piotr Sawicki, Marek Grzes, and Daniel G. Brown. 2022. Crowd score: A method for the evaluation of jokes using large language model ai voters as judges. *Preprint*, arXiv:2212.11214.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. *Preprint*, arXiv:2309.06180.

Rishabh Misra and Prahal Arora. 2023. Sarcasm detection using news headlines dataset. *AI Open*, 4:13–18.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Zen Simone and Cameron Cruz. 2020. Lmaonet – lstm model for automated objective humor scoring and joke generation. Cs224n custom project report, Stanford University. PDF available at https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/custom/15791516.pdf.

Leonard Tang, Alexander Cai, and Jason Wang. 2024. The naughtyformer: A transformer understands and moderates adult humor (student abstract). *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(13):16348–16349.

Orion Weller and Kevin Seppi. 2019. Humor detection: A transformer gets the last laugh. *"Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing"*.

Orion Weller and Kevin Seppi. 2020. The r/jokes dataset: a large scale humor collection. *"Proceedings of the 2020 Conference of Language Resources and Evaluation"*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

A Data Source of GRPO Trainer

Dataset	Source	Count
English		
Headlines	pnadel/nyt_headlines	2000
Word pairs	Randomly generated	2000

Table 3: Languages and Sources of Data used in RL Training.

B Prompt for Generating Joke Scores

The following prompt and its variances (translated to the respective languages) are used to generate scores for jokes :

> Observational jokes are an examination of everyday things or situations through a comedic lens. They cover topics familiar to almost everyone, even the most trivial aspects of life. Anecdotal humor, however, is pulled from the comedian's personal life and is popular with audiences because they can identify with their stories. You are a person who enjoys observational and anecdotal humour, as well as one-liners and irony. You appreciate a funny joke, but it isn't too easy to make you laugh, either. Your task is to rate a joke on a scale from 0 to 10, where 0 means it is not funny at all, and 10 means it is really hilarious. A mediocre joke typically gets a 5. A 9 or 10 score is very rare, and reserved for the best jokes only. Therefore, 8 is considered a very good rating, and you shouldn't be too generous with it. You should return only a valid JSON with the fields ‘rating‘ (that contains your rating, as an integer), and ‘reason‘, which justifies your answer. The joke is: {joke}

This prompt uses the idea of (Goes et al., 2022) to evaluate a joke from different perspectives, but we use different joke "types". Specifically, we use a list with the 10 most popular types of jokes from Masterclass. [7] Then, we extract those joke types that apply to text-only humor, and we include parts of their descriptions. We ask for a JSON-formatted response to easily extract the labels, and we ask to justify the ratings (the ‘reason‘ field) in hopes that this leads to more thoughtful responses.

[7] https://www.masterclass.com/articles/the-10-most-popular-types-of-jokes

C Data Source of Funniness Classifier

Language	Source	Count
English	Reddit	30,000
English	Synthetic	20,000
English	Naughtyformer	1,844
English	HumorDetection-Pun	4,216
English	Sarcastic Headline	26,607
Spanish	HAHA2019	26,419
Spanish	Chistes	2,420
Chinese	chinese-joke	16,100
Chinese	CFunSet	34,510
Total		**154754**

Table 4: Languages and Sources of Data used in Funniness Classifier Training

D Label Distribution for Funniness Classifier

Score	Number of Items
0	14,588
1	6,691
2	15,820
3	6,737
4	7,204
5	2,933
6	21,325
7	23,008
8	54,026
9	2,414
10	8
Total	154754

Table 5: Distribution for Each Label (0–10) in the in Funniness Classifier Training