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 an LLM serving as the policy and a custom joke rating model providing a 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 (Castro et al., 2026). 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 crowdsourced 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 approach, our system generates consistently humorous text from a given word pair or headline while remaining computationally efficient enough to be reproduced on a single academic-grade GPU.

1.1 SemEval Official Results

The output generated by our system ranks sixth in Task A: Joke in English, eighth in Task A: Joke in Spanish, and fifth in Task A: Joke in Chinese.

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

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 463–474

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**, 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 **Qwen2.5-7B-Instruct**, which fits on academic grade GPUs such as the NVIDIA L40S or A100.[3] Qwen-2.5-7B-Instruct is a 6.5B parameter, instruction-tuned causal language model, which uses a transformer architecture with 28 layers and Grouped-Query Attention (GQA) (Ainslie et al., 2023). The attention component leverages 28 Query heads and 4 Key-Value heads to support a context length of 32,768 tokens (Yang et al., 2025).

2.1.2 Training Data

To train the Qwen2.5-based model, we use 5000 news headlines from published datasets and 5000 randomly-generated word pairs for each language.[4] Details of the data mix and sources are provided in Appendix E.

2.2 Funniness Classifier

We are inspired by the prior work of Goes et al. (2022), showing that prompting LLMs (in this case GPT-3) 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 its predictions as a reward signal. This approach not only improves efficiency (and latency, by not having an LLM in the loop) but also increases determinism and reproducibility, as the learned model provides more stable and consistent reward values.

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.

2.2.1 Model Design

We train a Hierarchical Classifier architecture based on the XLM-RoBERTa-large encoder (Conneau et al., 2020) 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.

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

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

[3]Specifically, we use the latest checkpoint available at https://huggingface.co/Qwen/Qwen2.5-7B-Instruct at the time of writing.

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

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. 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. We use English jokes from the r/Jokes Dataset (Weller and Seppi, 2020) and humorous texts provided by Tang et al. (2024), Weller and Seppi (2019), and Misra and Arora (2023). For Spanish, we incorporate the HAHA 2019 corpus (Chiruzzo et al., 2020). For Chinese, we utilize the CFunSet (Yu et al., 2025). Additionally, we synthetically generate 20,000 additional English jokes for underrepresented scores. A detailed data breakdown is provided in Appendix B. To label the jokes, we use Llama-3-70B (Grattafiori et al., 2024) and GPT-OSS-120B (Agarwal et al., 2025) for English and Spanish jokes, and an uncensored Qwen-2.5-7B model (Yang et al., 2025) and DeepSeek-V3.1 (Shao et al., 2024) for Chinese. The prompt we use is included in Appendix A, and the distribution of the LLM-generated ratings is also provided in 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 straightforward heuristics.

1) The **formatting reward** ensures that the output is a valid, single joke by penalizing commentary or conversational artifacts, specific prefixes or 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.[5]

[5]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.

2) Because, over time, the policy tends to produce only a few high-scoring joke structures (such as "Why X? Because Y!"), we add a **structure diversity reward** that promotes the generation of diverse joke structures. We first classify each generated joke into one of eight structure categories: why-did, where-did, how-did, what-do-you-call, knock-knock, qa-punchline, observation, and one-liner, using simple regex-based pattern matching. We then maintain a sliding window of the 30 most recently generated structures and compute, for each category, its frequency within that window. The reward for a given joke is defined as $r_{\text{div}} = f_{\text{target}} - f_{\text{actual}}$, where $f_{\text{target}} = 1/k$ is the uniform target frequency over k observed structure types and f_{actual} is the current relative frequency of the joke's structure. This results in a positive signal for underrepresented structures and a negative signal for overrepresented ones, hence encouraging the policy to generate a broader range of joke structures rather than exploiting a few high-reward templates.

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** acts as a lightweight format constraint for the inputs containing headlines, penalizing outputs that exceed the target length or contain conversational artifacts (e.g., repeating the word "headline"), thereby keeping the model's generations concise and on-task.

4) Finally, a **coherence penalty** serves as an estimation for output quality by penalizing completions with an unusually high ratio of capitalized words (above 20%), which we found to correlate with incoherent outputs dominated by proper nouns or technical jargon.

As a consequence, the final scalar reward R is computed as a weighted sum of the five signals, $R = \sum_i w_i r_i$, where w_i are predefined reward weights. The ranges for each reward and their respective weights are shown in table 2.

3 Experimental Setup

3.1 Funniness Classifier

The XLM-RoBERTa joke rating model is fine-tuned for 100 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

Reward	Weight	Effective Range
`roberta_score`	1.0	$[0, 10]$
`structure_diversity`	1.5	$\approx [-1.5, 1.5]$
`word_pair_adherence`	2.0	$[-4, 4]$
`formatting`	0.5	$[-2.5, 0.5]$
`length_penalty`	0.5	$[-1, 0]$
`headline_adherence`	2.0	$[-2, 2]$
`coherence_penalty`	0.5	$\approx [-0.25, 0]$

Table 2: Weighted contribution range of each reward component to the total RL objective.

32 and a learning rate of 5×10^{-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 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 1 epoch 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 0.8. We use gradient accumulation over 8 steps with a per-device batch size of 1, resulting in an effective batch size of 8 for training. At each step, we sample 4 completions per prompt (corresponding to G in the original GRPO paper (Shao et al., 2024)), with a generation batch size of 4. The model is trained in bfloat16 precision with gradient checkpointing enabled to reduce memory consumption.

4 Results

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

4.1 Funniness Classifier

To select an appropriate reward model for scoring the joke quality in our training pipeline, we conduct an ablation study comparing two multilingual transformer encoder models: XLM-RoBERTa-Large (Conneau et al., 2020) (560M parameters) and mDeBERTa-v3-Base (He et al., 2023) (280M parameters). We compare two training regimes: (1) frozen backbone with trainable classification head and (2) full fine-tuning (FT). We evaluate the

Lang.	Model	Acc.	MAE
EN			
	Random Baseline	0.091	3.577
	Majority Baseline	0.295	2.859
	XLM-R-Large	0.390	1.796
	XLM-R-Large (FT)	**0.450**	**1.395**
	mDeBERTa-Base	0.373	1.987
	mDeBERTa-Base (FT)	0.445	1.450
ZH			
	Random Baseline	0.097	3.712
	Majority Baseline	0.457	2.095
	XLM-R-Large	0.512	1.715
	XLM-R-Large (FT)	0.517	**1.316**
	mDeBERTa-Base	0.493	1.853
	mDeBERTa-Base (FT)	**0.539**	1.377
ES			
	Random Baseline	0.134	3.238
	Majority Baseline	0.548	2.283
	XLM-R-Large	0.673	1.408
	XLM-R-Large (FT)	**0.706**	**1.108**
	mDeBERTa-Base	0.647	1.574
	mDeBERTa-Base (FT)	0.681	1.165

Table 3: Reward model ablation results compared against Random and Majority Class baselines. (FT) indicates full fine-tuning. Majority Baseline consistently predicts the most frequent label (8). For MAE, lower is better; for Accuracy, higher is better. Best scores within a language are bolded.

test splits of the joke datasets that we prepared for English, Chinese, and Spanish. We measure classification accuracy and Mean Absolute Error (MAE) on an 11-class ordinal scale (0–10).

To establish a performance floor for our reward model, we compare our results against two naive baselines: a Random Baseline, which predicts labels uniformly at random, and a Majority Baseline, which always predicts the most frequent label in our training set (label 8). As shown in Table 3, all transformer models outperform these baselines across every language. It further demonstrates that full fine-tuning is essential as it improves MAE on average by 22.4% (XLM-RoBERTa) and 24.7% (mDeBERTa) compared to training only a classifier. XLM-RoBERTa achieves consistently lower MAE across all languages (despite slightly lower accuracy on Chinese). For our training procedure, MAE is more relevant because the reward model must provide ordinal scores that correctly rank joke quality; lower prediction error yields more stable policy gradients during GRPO. Finally, using a single architecture (XLM-RoBERTa-Large) across all languages allows us to use a unified pipeline and expand to additional languages more easily. Consequently, we adopt fully fine-tuned XLM-RoBERTa-Large as our reward model for all three languages.

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 better in our subjective opinions.

4.2.1 English

The English model shows initial strong improvement, with total reward increasing from approximately 3.4 to a peak of 7.3 as illustrated in figure 1. However, we observe notable volatility throughout training and a decline toward the end, with final rewards settling around 6.7. The funniness score similarly peaks at 6.9 before declining to 6.7. Despite this instability, we find that the trained model generally produces jokes that are more funny and better at satisfying task constraints compared to the pre-GRPO policy and larger models such as ChatGPT. These findings are, inevitably, subjective and readers may draw their own conclusions based on the sample outputs provided in Appendix G.

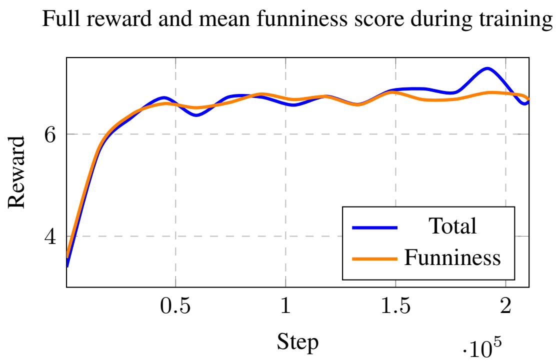

Figure 1: Training of English classifier. This plot shows the full training reward (blue) and the reward provided by the funniness classifier (orange) over the training steps (15-step rolling mean).

4.2.2 Spanish

The training progress of the Spanish model suggests the most successful training among the three languages, as illustrated in figure 2. The total reward increases steadily from approximately 3.1 to 7.5 over the course of training, while the funniness score improves from 4.4 to 7.3. Notably, the Spanish model achieves the highest final funniness scores and exhibits greater stability compared to both English and Chinese models. Subjectively, we observe that the Spanish jokes exhibit strong adherence to the task constraints, with many generations demonstrating genuine humor. Sample outputs are included in Appendix G.

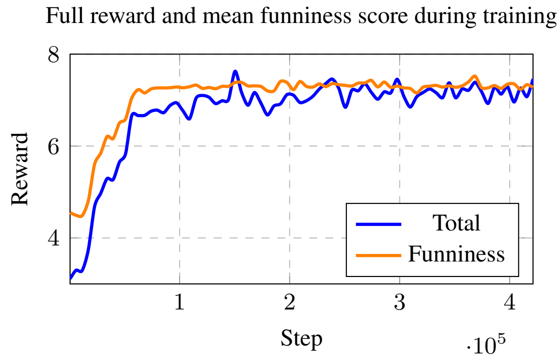

Figure 2: Training of Spanish classifier. This plot shows the full training reward (blue) and the reward provided by the funniness classifier (orange) over the training steps (15-step rolling mean).

4.2.3 Chinese

The Chinese model training presents the most challenging results, as shown in figure 3. While the model shows initial improvement, with total reward rising from approximately 5.2 to a peak of 6.5, we observe a concerning decline toward the end of training, with the final reward dropping to 5.8. Further, the funniness score plateaus at approximately 6.4, significantly lower than both the Spanish (7.3) and English (6.7) models. Hence, we consider the Chinese training to be only partially successful, warranting further investigation in future work. Potential causes for this will be explored in section 5.

4.3 Computational Efficiency

Our training approach has low computational requirements, making it accessible to academic researchers. Training was conducted on a single NVIDIA L40S GPU (48GB VRAM) using bfloat16 precision with gradient checkpointing. The English model completed training in 5.56 hours (210,599 steps), while the Spanish model required 9.71 hours (421,199 steps). The Chinese model trained for 390,451 steps with comparable runtime. Total training time across all three languages was approximately 15-16 hours on a single GPU. Memory consumption remained under 40GB during train-

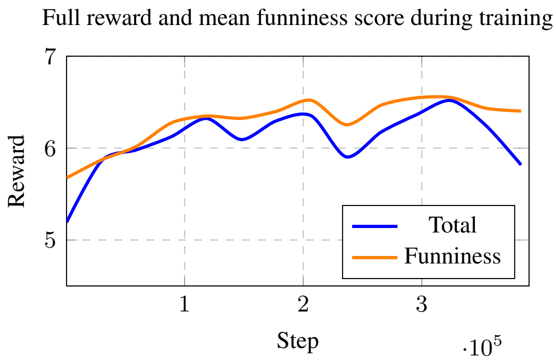

Figure 3: Training of Chinese classifier. This plot shows the full training reward (blue) and the reward provided by the funniness classifier (orange) over the training steps (15-step rolling mean).

ing, making this approach feasible for researchers with access to modern academic-grade GPUs.

5 Interpretation

The increase and subsequent convergence of the cumulative reward in the GRPO training plots for English and Spanish (1 and 2, respectively) provide 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). However, the aforementioned unsuccessful training of the Chinese joke generation model warrants caution. We suspect that this difficulty is a result of one or more of the following factors: a) the Chinese joke rater model provides less effective reward signals, b) the task of generating Chinese jokes may be inherently more challenging for this model architecture, or c) our hyperparameters and reward functions require further tuning for the Chinese language. Furthermore, the mixed use of Traditional and Simplified Chinese characters across datasets introduces additional uncertainty: the funniness classifier was trained exclusively on Simplified Chinese data, while the joke generation model was trained on a mix of both types of characters, and the final SemEval test data contains both. This script inconsistency likely contributes to the difficulty in training the Chinese model.

6 Conclusion

In this paper, we present the approach by Team TüLK for the SemEval-2026 Humor Generation task. We demonstrate that Group Relative Policy Optimization (GRPO), combined with a dedicated funniness classifier and heuristic reward signals, provides a computationally efficient framework for fine-tuning LLMs for joke generation under constrained tasks. Our system successfully generates humorous content in English and Spanish while adhering to strict word-pair and headline constraints, all within the resource limits of a single academic-grade GPU. However, the variance in training success across languages suggests that the quality of the reward signal remains the primary bottleneck. Future work will focus on refining multilingual reward modeling, perhaps by incorporating more diverse human-annotated datasets. Ultimately, our results show the potential of RL-based training in more subjective and creative linguistic domains.

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 method may not align perfectly 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. Additionally, the performance disparity observed across languages (i.e. the training instability in the Chinese model) indicates that the reward signal's effectiveness is dependent on the linguistic alignment and data quality of the initial classifier. Finally, because humor is inherently subjective and culturally charged, a scalar reward derived from a trained encoder may sometimes fail to capture subtle wordplay and context that define human humor.

Acknowledgements

We acknowledge support by the state of Baden-Württemberg through the bwHPC initiative and the use of the high-performance computing cluster bwUniCluster 2.0.

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.

Santiago Castro, Luis Chiruzzo, Santiago Góngora, Salar Rahili, Naihao Deng, Ignacio Sastre, Victoria Amoroso, Guillermo Rey, Aiala Rosá, Guillermo Moncecchi, J. A. Meaney, Juan José Prada, and Rada Mihalcea. 2026. SemEval-2026 Task 1: MWAHAHA, Models Write Automatic Humor And Humans Annotate. In *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026)*.

Luis Chiruzzo, Santiago Castro, and Aiala Rosá. 2020. HAHA 2019 Dataset: A Corpus for Humor Analysis in Spanish. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5106–5112, Marseille, France. European Language Resources Association.

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.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

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.

Zhenghan Yu, Xinyu Hu, and Xiaojun Wan. 2025. CFunModel: A "Funny" Language Model Capable of Chinese Humor Generation and Processing. *Preprint*, arXiv:2503.20417.

A Prompt for Generating Joke Scores

The following prompt and its translations 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.[6] Then, we extract those joke types that apply to text-only humor and 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.

B Data Sources for Funniness Classifier

In this section, we outline the sources of the jokes used to train our reward model. As shown in Table 4, our data mix consists of existing academic datasets, Reddit-based collections, and additional synthetically generated humor.

C Labeling Methods for Funniness Classifier

To provide a scalar reward signal, the unlabeled jokes were processed in various ways to obtain funniness ratings. Table 5 summarizes the specific approaches used for each language, leveraging an ensemble approach where possible.

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

Language	Source	Count
English	one-million-reddit-jokes	40,000
English	LLM-generated	20,000
English	Sarcastic Headlines	32,616
English	Short-jokes	19,987
English	Humorous Oneliners	5,251
	Subtotal	*117,854*
Spanish	HAHA2019	24,000
Spanish	CHISTES	2,419
	Subtotal	*26,419*
Chinese	chinese-joke	15,732
Chinese	CFunSet	34,517
	Subtotal	*50,249*
Total		**194,522**

Table 4: Dataset sources of funniness-classifier training data in each language

Language	Labeling Method	Samples
English	Human + Llama 3.1	117,854
Chinese	Qwen + DeepSeek	50,249
Spanish	Llama 3.1	26,419

Table 5: Labeling methods of funniness-classifier training data in each language

D Label Distribution for Funniness Classifier

The resulting distribution of the 0–10 funniness scale is presented in Table 6. We observe that the Spanish labeling via Llama 3.1 shows a more binary behavior compared to the more distributed English dataset.

E Data Sources for GRPO Training

The prompts used during the reinforcement learning phase come from a combination of real-world news headlines and synthetically generated word pairs. Table 7 outlines the specific sources used to create the 30,000 unique prompts used in the GRPO training.

F Prompt Templates for GRPO Training

We use a standardized set of prompts for both word-pair and headline-based tasks across all languages. The exact templates provided to the Qwen-2.5 policy are listed in Table 8.

Label	English		Chinese		Spanish	
	Count	**%**	**Count**	**%**	**Count**	**%**
0	13,417	11.4	8,130	16.2	1,169	4.4
1	6,691	5.7	284	0.6	—	—
2	8,344	7.1	1,247	2.5	7,365	27.9
3	6,701	5.7	871	1.7	—	—
4	7,182	6.1	1,517	3.0	14	0.1
5	2,793	2.4	1,175	2.3	140	0.5
6	17,778	15.1	2,549	5.1	3,384	12.8
7	12,628	10.7	5,753	11.4	24	0.1
8	34,721	29.5	22,996	45.8	14,319	54.2
9	2,340	2.0	5,646	11.2	4	0.0
10	5,259	4.5	81	0.2	—	—
Total	**117,854**	**100.0**	**50,249**	**100.0**	**26,419**	**100.0**

Table 6: Label distribution across languages on a 0–10 funniness scale. English data uses human annotations with balanced distribution (mode: 8, 29.5%). Chinese data from LLM ensemble shows strong skew toward label 8 (45.8%). Spanish data from Llama 3.1 exhibits bimodal distribution at labels 2 (27.9%) and 8 (54.2%) with several labels absent.

Language	Dataset Source	Count
English		
	Headlines (vblagoje/cc_news)	5,000
	Word pairs (randomly generated)	5,000
	Subtotal	*10,000*
Spanish		
	Headlines (hacktoberfest-corpus-es/colmbian_spanish_news)	5,000
	Word pairs (randomly generated)	5,000
	Subtotal	*10,000*
Chinese		
	Headlines (jed351/rthk_news)	5,000
	Word pairs (randomly generated)	5,000
	Subtotal	*10,000*
Total		**30,000**

Table 7: Data sources used in reinforcement learning training across three languages.

Language	Task Type	Prompt Template
English	Word Pair	Generate a funny joke using these two words: '{w1}', '{w2}'. Only respond with the joke and nothing else.
English	Headline	Generate a funny joke related to this headline: '{headline}'. Only respond with the joke and nothing else.
Chinese	Word Pair	用这两个词生成一个有趣的笑话:'{w1}'、'{w2}'。只回复笑话,不要回复其他内容。
Chinese	Headline	根据这个标题生成一个有趣的笑话:'{headline}'。只回复笑话,不要回复其他内容。
Spanish	Word Pair	Genera un chiste gracioso usando estas dos palabras: '{w1}', '{w2}'. Solo responde con el chiste y nada más.
Spanish	Headline	Genera un chiste gracioso relacionado con este titular: '{headline}'. Solo responde con el chiste y nada más.

Table 8: Prompt templates used for joke generation.

G Example Outputs

Finally, in this section, we provide examples of the jokes generated by our system compared to the base model and commercial baselines. Tables 9 through 14 show how our fine-tuned policy handles specific constraints like word integration and headline relevance in English, Chinese, and Spanish.

Task Type	Task Detail
Word Pair	Hammer, Flower
Model	**Output**
Our joke generator	A hammer went to a flower and said, “You’re looking a bit wilted, do you need some nail polish?”
Qwen2.5-7B-Instruct	Why did the flower need a hammer? To beat the heat and stay cool!
ChatGPT	I tried to fix my garden with a hammer and a flower—turns out one helps things grow, and the other just helps me apologize to the flower.

Table 9: Sample output for English word pair. In all sample output tables, Qwen2.5-7B-Instruct refers to the model before training; ChatGPT refers to the product available at the time of writing.

Task Type	Task Detail
Headline	Doing these fall garden chores will make your spring easier
Model	**Output**
Our joke generator	Raking leaves is like nature’s way of telling you to buy a leaf blower, but if you ignore the message, your allergies will thank you in the spring.
Qwen2.5-7B-Instruct	Why did the gardener wait until fall to organize his garden tools? Because he heard spring was a tool!
ChatGPT	I did all my fall garden chores to make spring easier—now spring just arrives earlier to watch me ignore them again.

Table 10: Sample output for English headline

Task Type	Task Detail
Word Pair	摇晃, 椅子
Model	**Output**
Our joke generator	椅子突然摇晃着对地板说：“你最近压力好大啊！”地板苦笑道：“别提了，我家椅子每天都跟我一起摇晃！”
Qwen2.5-7B-Instruct	为什么那把椅子总是摇头晃脑？因为它听说这样更容易吸引“靠”它的朋友！
ChatGPT	我一坐到那把会摇晃的椅子上就开始反思人生——原来不是我不稳定，是椅子在替我诚实表达内心状态。

Table 11: Sample output for Chinese word pair.

Task Type	Task Detail
Headline	马斯克的xAI洽融150亿美元融资
Model	**Output**
Our joke generator	马斯克：我已经为xAI洽融了150亿美元。机器人：你需要我做什么？马斯克：你去把我那150亿给我取回来吧！
Qwen2.5-7B-Instruct	为什么马斯克的xAI要融资150亿美元？因为它听说宇宙里99%的星星都是融资用的！
ChatGPT	听说马斯克的xAI融了150亿美元——原来AI不是为了拯救人类，是为了终于能在我问“今天吃什么”时不再回答“嗯...正在思考中”。

Table 12: Sample output for Chinese headline

Task Type	**Task Detail**
Word Pair	lavar, bicicleta
Model	**Output**
Our joke generator	Un día, mi bicicleta se ensució, así que decidí lavarla. Resultó ser una experiencia más agotadora que subir al Everest.
Qwen2.5-7B-Instruct	¿Por qué la bicicleta necesitaba ir al taller de lavado? Because she was tired of pedaling through a sea of sweat!
ChatGPT	Quise lavar mi bicicleta y ahora parece nueva… lástima que yo sigo igual de cansado.

Table 13: Sample output for Spanish word pair.

Task Type	**Task Detail**
Headline	Video Así han cambiado los actores de “Stranger Things” entre la primera y la última temporada
Model	**Output**
Our joke generator	El actor que interpretaba al monstruo ha crecido tanto que ahora es el productor ejecutivo de la serie.
Qwen2.5-7B-Instruct	¿Por qué los actores de "Stranger Things" necesitaban tanto maquillaje en la última temporada? Porque la diferencia entre la primera y la última temporada es como la delgada línea que separa el umbral de la madurez... y un charco de gomitas Skittles.
ChatGPT	Ver cómo han cambiado los actores de “Stranger Things” es como revisar tus fotos de la secundaria: todos crecieron, menos tus excusas para no hacer nada.

Table 14: Sample output for Spanish headline