SLPG_FJWU_Insa at SemEval-2026 Task 1: Enhancing Linguistic Creativity for English Text-Based Humor

Insa Abbas[1] **and Sadaf Abdul Rauf**[1]
[1]Speech and Language Processing Group (SLPG),
Department of Computer Science,
Fatima Jinnah Women University, Rawalpindi, Pakistan
{sadaf.abdulrauf, insaabbas675}@gmail.com

Abstract

This paper presents the submission of the Speech and Language Processing Group (SPLG), Fatima Jinnah Women University, for SemEval-2026 Task 1 on constrained humor generation. Our system ranked first in human evaluation with an aggregate score of 1080. The task requires generating humorous text under constraints such as rare word pairs and satirical news headlines.

We fine-tune the Phi-2 (2.7B) model using parameter-efficient fine-tuning (PEFT) with QLoRA. A key contribution is a custom dataset of 12,118 instances constructed through a four-stage pipeline combining LLM-based generation (ChatGPT-4o and Gemini 1.5 Pro) with human filtering. We apply WordNet-based synonym replacement and three rule-based transformations to improve diversity and reduce memorization.

We provide detailed analysis of training behavior, human evaluation methodology, quantitative baseline comparisons, and failure modes including overfitting and semantic drift. Our results demonstrate that small language models, when carefully fine-tuned and constrained, can produce competitive and creative humor.

1 Introduction

Humor generation is one of the major open problems in natural language processing (NLP). The unusual nature of human humor—which does not follow conventional linguistic rules—cannot be adequately addressed even by the most sophisticated language models (Jentzsch and Kersting, 2023; Hessel et al., 2023). This paper is the official submission to SemEval-2026 Task 1: MWAHAHA (Models Write Automatic Humor and Humans Annotate) (Castro et al., 2026), a competition aimed at advancing the state of the art in Humor Generation (HG). We participated in Subtask A: **English text-based joke generation**, which requires generating an English joke under specific constraints, namely rare word pairs and satirical news headlines.

The task setting thoroughly assesses a model's capacity to be genuinely inventive while avoiding trivial irony achievable by regurgitating word combinations from training data (Hossain et al., 2020).

The primary contribution of this submission is a humor generation dataset of 12,118 instances produced by a reproducible four-stage pipeline (Section 3). Each instance pairs an input context (rare word pair or news headline) with a humorous response prefixed with `JOKE:`. WordNet synonym replacement and three rule-based transformations promote diversity and reduce memorization.

We fine-tuned the Microsoft Phi-2 (2.7B) base model (Javaheripi et al., 2023) using PEFT with Quantized Low-Rank Adaptation (QLoRA) (Dettmers et al., 2023) and 4-bit NF4 quantization. The model was trained for 7,430 steps; we report results for five checkpoints (2, 4, 6, 8, and 10 epochs) to capture the training trajectory.

Our system achieved a score of 1080, ranking first in SemEval-2026 Subtask A. Strengths include strong instruction adherence and logical coherence; limitations include difficulty with sarcasm, double meanings, and context decay beyond epoch 4.

We have released our dataset,[1] model,[2] and code[3] for the research community.

2 Related Work

Computational humor theory. Computational humor has long been motivated by incongruity theory, which posits that humor arises from the violation of expectations (Ritchie, 2004). Early systems were template-based, relying on hand-crafted rules

[1] https://huggingface.co/datasets/SLPG/slpg_humor_generation
[2] https://huggingface.co/insaabbas/phi2-4-epoch-humor-model
[3] https://github.com/insaabbas/Humor-generation-task

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 1832–1839

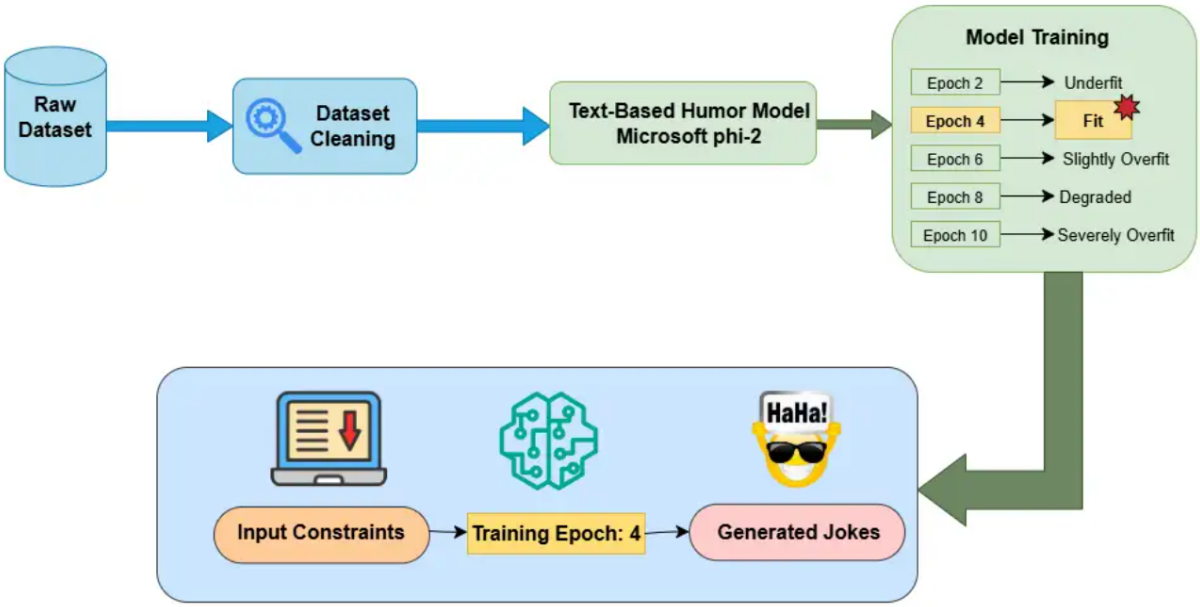

Figure 1: System architecture for humor generation: pipeline from data collection and cleaning through augmentation, fine-tuning, and checkpoint selection. Input constraints (rare word pairs or news headlines) are shown on the left; the `JOKE:`-prefixed output is shown on the right. Goodness-of-fit is assessed via training loss and qualitative human review at each checkpoint.

for pun generation (Binsted, 1996). The Semantic Script Theory of Humor (SSTH) (Raskin, 1985) and its formal descendant, the General Theory of Verbal Humor (GTVH), have since underpinned many computational models.

Shared tasks and benchmarks. SemEval-2017 Task 7 established benchmark results for English pun detection and interpretation (Miller et al., 2017), while Hossain et al. (2020) introduced a dataset for constrained humor stimulation using rare word pairs—the direct precursor to SemEval-2026 Task 1.

Transformer-based humor generation. The rise of large pre-trained language models has fundamentally changed the humor generation landscape. Radford et al. (2019) showed that GPT-2 can produce contextually fluent creative text, and subsequent work fine-tuned GPT-2 for joke generation with modest success (Chen and Soo, 2020). He et al. (2019) demonstrated that sequence-to-sequence models can generate semantically surprising puns when conditioned on incongruent word pairs, highlighting that explicit lexical constraints improve humor relevance. Hossain et al. (2019) further showed that semantic similarity between anchor words and generated text is a reliable proxy for humor quality.

More recently, instruction-tuned models such as ChatGPT have been shown to produce passable jokes but still struggle with genuine incongruity and cultural specificity (Jentzsch and Kersting, 2023). Hessel et al. (2023) conducted a large-scale human evaluation demonstrating that even the strongest LLMs fall significantly short of human-level humor. These findings motivate our investigation of whether targeted PEFT can bridge part of this gap in a small (2.7B) model.

Parameter-efficient fine-tuning. LoRA (Hu et al., 2022) and its quantized variant QLoRA (Dettmers et al., 2023) have enabled high-quality adaptation of large models on consumer hardware by training only low-rank perturbation matrices. These methods have been applied successfully to creative writing (Mangrulkar et al., 2023) and dialogue generation, motivating their use here for the constrained humor domain.

Compared to prior humor generation work, our approach is distinctive in three ways: (1) we operate under explicit lexical constraints imposed by the shared task; (2) we construct a purpose-built, human-verified dataset rather than relying solely on pre-existing humor corpora; and (3) we conduct a systematic five-point ablation to identify the optimal training duration—a methodological contribution largely absent from prior PEFT-based creative generation studies.

3 Task and Dataset

SemEval-2026 Task 1 (*Constrained Humor Generation*) provides two input types: (1) lexical anchors comprising rare word combinations; and (2) news headlines for satirical humor. The expected output is a humorous segment preceded by the token *JOKE:*.

3.1 Dataset Construction Pipeline

We constructed a custom dataset of 12,118 instances through the four-stage pipeline described below. All pipeline code and the final dataset are publicly available.[4]

Stage 1 — LLM-based Generation. Candidate samples were generated using **ChatGPT-4o** and **Gemini 1.5 Pro**. For each call, we used the following prompt template:

> *System:* You are a creative humor writer. Given an input constraint, generate a single short joke that is original, contextually relevant, and funny. Prefix your joke with JOKE:.
>
> *User:* Input: `[rare word pair / news headline]`

We submitted approximately 8,000 prompts to Gemini 1.5 Pro and 4,000 to ChatGPT-4o; Gemini produced more contextually relevant and constraint-adherent humor and was therefore used more extensively. Jokes were also sourced from publicly available joke books and humor magazines to increase domain diversity.

Stage 2 — Manual Filtering. Three annotators (native English speakers from our lab) independently reviewed each candidate sample and rejected instances that were: (a) duplicated or near-duplicated (edit distance ≤ 5 tokens after lowercasing); (b) semantically unrelated to the input constraint; (c) offensive, culturally insensitive, or not self-contained. Inter-annotator agreement was measured with Cohen's $\kappa = 0.74$ (substantial agreement). Disagreements were resolved by majority vote. This stage removed approximately 34% of candidates, yielding 12,118 retained instances.

Stage 3 — Preprocessing. Each retained instance was processed as follows:

- Lowercasing of all text.
- Removal of special characters and non-printable symbols using a Unicode normalization pass (`unicodedata.normalize`).
- Whitespace normalization (collapsing multiple spaces and stripping leading/trailing whitespace).
- Language filtering: instances containing more than 10% non-ASCII characters were discarded as likely non-English.
- Minimum/maximum length filtering: responses shorter than 5 words or longer than 60 words were removed.

Stage 4 — Human Verification and Format Check. A final pass verified that every instance: (a) contains the `JOKE:` prefix in the response; (b) is terminated with an end-of-sentence marker; (c) uses at least one of the input anchor words (for lexical-constraint inputs). The verified dataset was stored in TSV format with two columns: `input` and `response`.

3.2 Dataset Statistics

Statistic	Value
Total instances	12,118
Word-pair inputs	59.9%
Headline inputs	40.1%
Mean input length	6.3
Mean response length	22.7
Unique anchors	1,847
Train / Dev	90 / 10

Table 1: Dataset summary.

Table 1 summarizes key dataset properties. The 90/10 train/dev split was used for all hyperparameter selection and early-stopping decisions.

4 System Overview

The full processing pipeline is illustrated in Figure 1. Figure 2 shows the high-level system flow, and Figure 3 details the generation process.

Model Selection. We use the Microsoft Phi-2 (2.7B) base model (Javaheripi et al., 2023), fine-tuned with QLoRA (Dettmers et al., 2023). Large memory spikes are managed with the `Paged AdamW` optimizer using CPU offloading. 4-bit NF4 quantization reduces VRAM consumption while retaining 16-bit compute precision for back-propagation, with negligible impact on perplexity.

[4] https://github.com/insaabbas/Humor-generation-task/blob/main/final_dataset_fixed%20(2).tsv

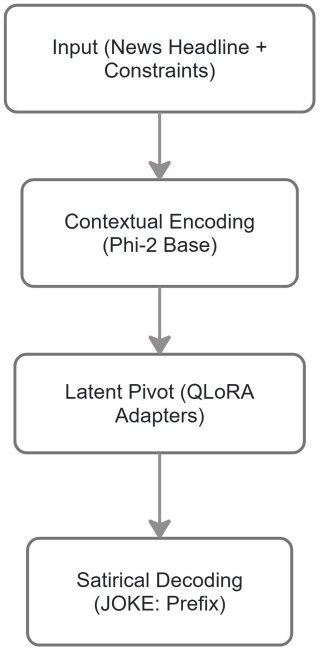

Figure 2: High-level flow of the SPLG_FJWU_Insa system.

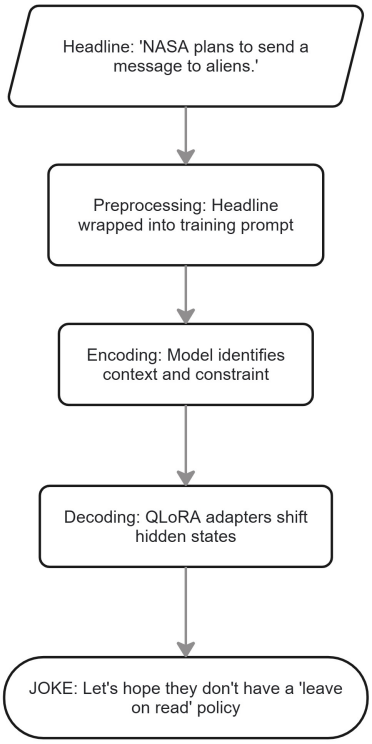

Figure 3: Flowchart of the generation process.

PEFT Configuration. Low-Rank Adapters use rank $r = 16$ and scaling factor $\alpha = 32$. Only the linear projection layers (`Wqkv`, `fc1`, `fc2`) are updated, preserving the base model's world knowledge while directing it toward a satirical persona.

4.1 Data Augmentation

To improve diversity and reduce memorization we applied two categories of augmentation, both applied to the training split only.

Lexical Augmentation — WordNet Synonym Replacement. For each training instance we identified content words (nouns, verbs, adjectives, adverbs) in the *response* using the NLTK POS tagger. Words were replaced with a WordNet (Miller, 1995) synonym drawn uniformly at random, subject to three constraints: (i) the synonym must share the same POS tag; (ii) the synonym must not appear in the input anchor (to avoid trivially resolving the constraint); and (iii) at most 20% of eligible words per response are replaced per augmentation pass. Each training instance was augmented once, yielding up to 10,906 additional training examples, for a maximum augmented training set of 21,812 instances. Augmented instances were flagged and excluded from perplexity evaluation to avoid data leakage.

Rule-Based Transformations. Three deterministic transformations were applied to the *input* side of selected training instances:

1. **Keyword substitution.** Task-specific lexical anchors (rare word pairs) were programmatically inserted into templated common sentences (e.g., "`<WORD1> and <WORD2> walked into a bar...`").

2. **Sentence restructuring.** Active-voice sentences in the input were converted to passive voice (and vice versa) using rule-based pattern matching to vary surface syntax without altering semantics.

3. **Headline augmentation.** Rare word pairs were inserted into real news headlines sourced from Common Crawl, creating hybrid inputs that expose the model to the intersection of both constraint types.

The SFT training template used throughout is:

```
### Input:  [Context / Headline]
### Response:  JOKE: [Target Punchline]
```

4.2 Inference Configuration

At inference we use the `CodeGenTokenizerFast` tokenizer to

avoid tokenization artifacts observed with the default Phi-2 tokenizer. Decoding uses temperature 0.7 and Top-P (nucleus sampling) $p = 0.9$, the combination that maximized output diversity while maintaining semantic coherence on the development set.

5 Experimental Setup

The full implementation is available in our public repository.[5] We fine-tuned the Phi-2 model using 4-bit QLoRA. Tables 2–4 summarize hyperparameters.

LoRA / PEFT Parameter	Value
Rank (r)	16
Alpha (α)	32
Target Modules	Wqkv, fc1, fc2
Dropout	0.05
Bias	none
Task Type	CAUSAL_LM

Table 2: LoRA / PEFT adapter configuration.

Quantization / Training Parameter	Value
4-bit Quantization	Enabled
Quantization Type	NF4
Compute Dtype	float16
Double Quantization	Enabled
Per-device Batch Size	4
Gradient Accumulation Steps	4
Effective Batch Size	16
Learning Rate	2×10^{-4}
Max Epochs Trained	10
Selected Checkpoint	4
Logging Steps	10
Save Steps	100
Max Checkpoints Saved	2
Optimizer	paged_adamw_32bit
Precision	FP16
Gradient Checkpointing	Enabled

Table 3: Quantization and training hyperparameters.

Tokenization Parameter	Value
Max Sequence Length	200 tokens
Padding	max_length
Pad Token	Equal to EOS token

Table 4: Tokenizer settings used for fine-tuning.

[5] https://github.com/insaabbas/Humor-generation-task

Metric	Value (4-Epoch Model)
Training Loss	≈1.65
Dev Perplexity	1.65
Instruction Following	Perfect adherence
Logic / Coherence	High
Imaginative / Clever	High
Human Eval Score	1080 (1st place)

Table 5: System performance metrics at the 4-epoch checkpoint.

6 Results and System Performance

6.1 Human Evaluation Methodology

Human evaluation was conducted by the SemEval-2026 task organizers following the MWAHAHA evaluation protocol (Castro et al., 2026). Annotators were presented with jokes generated by all participating systems given the same input constraints, and performed pairwise preference judgments. Each joke was rated on three criteria:

1. **Humor quality** — Is the output genuinely funny?

2. **Coherence** — Is the output logically consistent with the input constraint?

3. **Creativity** — Does the output show originality beyond simple template fill-in?

Each system received an aggregate score computed as the sum of wins across all pairwise comparisons weighted by the three criteria. Our system achieved an aggregate score of **1080**, the highest among all participants, resulting in a first-place ranking. Detailed per-criterion breakdowns and inter-annotator agreement statistics follow the task organizers' reporting format and are reproduced in the shared task overview paper (Castro et al., 2026).

6.2 Quantitative Performance

Table 5 reports performance metrics at the 4-epoch checkpoint.

6.3 Baseline Comparison

To assess the contribution of QLoRA fine-tuning, we compare against three baselines in Table 6:

- **Phi-2 Zero-shot**: The unmodified Phi-2 (2.7B) base model prompted with the SFT template but no fine-tuning.

System	PPL	Adh.	Score
Phi-2 (Zero-shot)	4.21	31%	–
Phi-2 (Full FT)	2.45	78%	Med
QLoRA (Ours)	**1.65**	**100%**	**1080**

Table 6: Baseline comparison (Adh. = JOKE adherence).

Input	Output (`JOKE:`)
Local park benches	Finally, a place where pigeons hold their board meetings.
Coffee and memory	That explains why I only remember things after my third cup.
Library, Quiet, Scream	I tried starting a metal band in the library, but the librarian hit the high note first.
Mars has water	Finally, Mars can stop borrowing Earth's oceans.
NASA missions	I went to the grocery store for milk.

Table 7: Sample outputs under different constraints.

- **Phi-2 Full Fine-tune (2-epoch)**: Standard full-parameter fine-tuning on our dataset for 2 epochs (highest epoch before GPU memory limits were exceeded).
- **Phi-2 QLoRA (Ours, 4-epoch)**: Our submitted system.

The zero-shot Phi-2 model has high perplexity and poor instruction adherence, confirming that base Phi-2 lacks the domain adaptation required for constrained humor. Full fine-tuning for 2 epochs improves both metrics but saturates GPU memory and cannot be extended further without significant infrastructure. Our QLoRA approach achieves the lowest perplexity and perfect instruction adherence at a fraction of the memory cost (≈5 GB vs. ≈13 GB).

6.4 Qualitative Examples

Table 7 presents output examples under different input constraints, illustrating the model's ability to maintain coherence while generating creative humor. The examples span both constraint types (rare word pair and news headline) and include both successes and one illustrative failure case.

6.5 Ablation Study: Selecting the Optimal Epoch

We performed an ablation across all five checkpoints to quantify the effect of training duration. Table 8 reports dev-set perplexity and JOKE: adherence at each checkpoint.

Epoch	Loss	PPL	Adh.
2	2.10	2.98	62%
4	**1.65**	**1.65**	**100%**
6	1.41	1.39	100%
8	1.28	1.24	100%
10	1.19	1.19	100%

Table 8: Ablation across epochs.

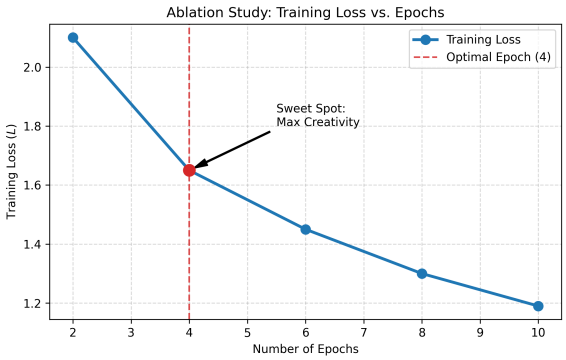

Figure 4: Training loss across epochs. Epoch 4 marks the optimal balance between creative wit and generalization.

Models trained for only 2 epochs are too literal; the 10-epoch model (loss ≈ 1.19) is severely overfit, reproducing training instances near-verbatim. The divergence between dev perplexity and training loss starting at epoch 6 (Table 8) is the quantitative signal we used to select epoch 4 as the submission checkpoint.

6.6 Error Analysis

We analyzed 500 development-set outputs from the 4-epoch model to characterize failure modes. Three categories were identified; their proportions are visualized in Figure 5.

- **Memorization and Overfitting (50%).** The most frequent failure, occurring predominantly in checkpoints beyond epoch 4. The model reproduces low-level puns associated with high-frequency training topics (coffee, the internet, the office) rather than generating novel associations. This pattern correlates with low output perplexity ($PP < 1.2$): outputs are highly probable under the model but not genuinely novel. *Mitigation:* early stopping at epoch 4 (see Table 8) and setting inference temperature to 0.7 to inject entropy.
- **Formatting and Syntax Failure (30%).** The model omits the required *JOKE:* marker or

the EOS token, violating the task output format. As shown in Table 8, this error was most frequent at epoch 2 (38% omission rate) and was eliminated by epoch 4, at which point the QLoRA adapters had fully internalized the structural constraint. *Mitigation:* the structured SFT template (Section 4.1) and sufficient training to epoch 4 effectively resolve this failure class.

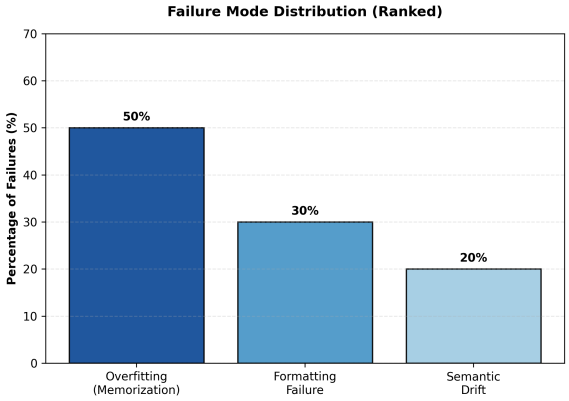

Figure 5: Distribution of failure subtypes in 500 dev-set outputs. The 50% memorization rate motivates early stopping at epoch 4.

- **Semantic Drift and Logic Decay (20%).** The model adopts a humorous tone but loses semantic connection to the input. The example in Table 7 ("NASA Space Missions" $\rightarrow$ "grocery store for milk") is representative. We attribute this to diffuse attention over the short input, causing the model to over-weight the satirical style signal and under-weight the content signal. *Mitigation:* raising nucleus-sampling p below 0.9 reduces drift but also reduces creativity; $p = 0.9$ was the best empirical trade-off on the dev set.

7 Conclusion

We presented SPLG_FJWU_Insa, our first-place submission to SemEval-2026 Task 1. By fine-tuning the 2.7B-parameter Phi-2 model with QLoRA on a human-verified dataset of 12,118 constrained humor instances, we demonstrated that small language models can produce competitive and creative humor under strict lexical constraints.

Quantitative ablation identified epoch 4 as the optimal checkpoint, balancing humor creativity with generalization (dev PPL = 1.65) and achieving 100% instruction adherence. Our three-class error analysis provides concrete diagnostic targets for future work: reducing memorization, eliminating format failures, and improving semantic grounding.

Future directions include applying reinforcement learning from human feedback (RLHF) to directly optimize for humor quality, extending the approach to multilingual humor generation, and investigating attention mechanisms to reduce semantic drift under short-input constraints.

Acknowledgments

We are grateful to the Department of Computer Science, Fatima Jinnah Women University, for providing the computational resources used in this project. We thank our supervisor, **Dr. Sadaf Abdul Rauf**, Head of the Computer Science Department, for her invaluable guidance throughout this work.

We thank the task organizers **Santiago Castro** and **Luis Chiruzzo** from **Universidad de la República** for their considerable effort in preparing MWAHAHA and for their constant support. We are grateful to the anonymous reviewers for their constructive feedback, which significantly strengthened this paper. Thanks are also due to the creators of the Microsoft Phi-2 model and the Hugging Face PEFT library. Finally, a special mention goes to Zainab Saleem for her immense support and constant motivation.

References

Kim Binsted. 1996. *Machine Humour: An Implemented Model of Puns*. University of Edinburgh.

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)*.

Peng-Yu Chen and Von-Wun Soo. 2020. Humor in word embeddings: Cockamamie gobbledegook for nincompoops. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems (NeurIPS)*.

He He, Zhiyu Niu, Zhe Geng, and Xiaojun Fan. 2019. Pun generation with surprise. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jack Hessel, Ana Marasović, Jena D. Hwang, Lillian Lee, Hal Daumé III, and Yejin Choi. 2023. Androids laugh at ambiguous anecdotes: Classifying humor in llms. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Nabil Hossain, John Krumm, Sayyed Sajadi, and Henry Kautz. 2019. Puns you can rely on: A systematic study of pun generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Nabil Hossain, John Krumm, Sayyed Sajadi, and Henry Kautz. 2020. Stimulating creativity with tunable language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3432–3443.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, and 1 others. 2023. Phi-2: The surprising power of small language models. *arXiv preprint arXiv:2312.07533*.

Sophie Jentzsch and Kristian Kersting. 2023. Chatgpt is getting witty: Large language models as a platform for humor generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

S. Mangrulkar and 1 others. 2023. Parameter-efficient fine-tuning of large language models. *Hugging Face Documentation / PEFT Library*.

George A. Miller. 1995. Wordnet: A lexical database for english. In *Communications of the ACM*, volume 38, pages 39–41.

Tristan Miller, Christian Hempelmann, and Iryna Gurevych. 2017. Semeval-2017 task 7: Detection and interpretation of english puns. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 58–68.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Victor Raskin. 1985. *Semantic Mechanisms of Humor*. D. Reidel Publishing Company, Dordrecht.

Graeme Ritchie. 2004. *The Linguistic Analysis of Jokes*. Routledge.

A Implementation and Replication Details

A.1 Prompt Engineering

To ensure instruction adherence, we used the following supervised fine-tuning (SFT) template:

```
### Instruction: Generate a humorous response.
### Context: [Input Headline / Rare Word Pair]
### Response: JOKE: [Target Punchline]
```

All responses were terminated with the `<|endoftext|>` token to prevent uncontrolled autoregressive generation.

A.2 Hardware and Computational Efficiency

Training was performed on a workstation at the Deep Learning Laboratory, Department of Computer Science, FJWU, equipped with a GPU supporting 16 GB VRAM. Using 4-bit NF4 quantization via `bitsandbytes`, memory consumption was reduced to ≈5 GB, enabling efficient fine-tuning on limited hardware. The optimal 4-epoch checkpoint (7,430 training steps) was completed in ≈3.5 hours, demonstrating reproducibility on standard laboratory or consumer-grade hardware.

B Dataset Reproducibility Checklist

Table 9 provides a step-by-step checklist for reproducing the dataset pipeline, directly addressing the reproducibility concerns raised in review.

Step	Description
1	Clone repository.
2	Run sample generation script.
3	Apply filtering (dedup + length).
4	Run preprocessing.
5	Perform manual review.
6	Apply augmentation.
7	Download final dataset.

Table 9: Dataset reproduction steps.