INF-rsrs at SemEval-2026 Task 1: Is the best really better? The limits of creative work in the era of LLMs

Guilherme T. Bazzo, Eduardo D. Faé, Júlia da Rocha Junqueira, Higor Moreira and **Lucas R. C. Pessutto**
Institute of Informatics – UFRGS – Porto Alegre – Brazil
{gtbazzo, edfae, julia.junqueira, hmoreira, lrcpessutto}@inf.ufrgs.br

Abstract

Generating humor is a complex and challenging task for Large Language Models (LLMs), requiring both linguistic creativity and strict adherence to constraints. This paper presents INF-rsrs, our solution for SemEval 2026 Task 1: Humor Generation, which tasks models with creating jokes from headlines and word pairs without labeled data. We propose a two-stage framework: a *production stage* and a *selection stage*. The production stage employs diverse model families and hyperparameter configurations to generate a wide range of candidate jokes, with each candidate generated by an LLM prompted in the role of a comedian under structured constraints to ensure relevance and humor. Our system was designed to substantiate our claim that the direct use of LLMs in creative works, such as humor generation, hits a hard ceiling that is inescapable through simple prompting. Our proposed system tied in first place in the task ranking, obtaining a top-tier performance.

1 Introduction

Humor is essential for human interaction and cognitive functioning, playing a fundamental role in facilitating social bonds (Savage et al., 2017). Its benefits range from reducing anxiety and emotional distress to improving learning outcomes (Buxman, 2008). But replicating human-like humor remains a significant challenge for artificial machines (Jentzsch and Kersting, 2023). Therefore, enhancing Large Language Models (LLMs) with a sense of humor has become a key task in natural language generation (Hessel et al., 2023; Sakabe et al., 2025; Romanowski et al., 2025).

SemEval 2026's Task 1 – Humor Generation (Castro et al., 2026), introduces a benchmark for evaluating humor generation. It is the first task dedicated to advancing the state of the art in computational humor generation. The challenge lies in generating a joke using a set of constraints, where each joke must contain two target keywords from a list or must be related to a specific news article headline. The task focuses on genuine generation rather than memorization.The evaluation was based on human preference judgments, using a pairwise comparison setup ('battle') where annotators choose the funnier joke under identical constraints.

In this paper, we propose a two-stage framework to address the task by combining multi-agent generation with an arena-based selection mechanism. Our approach targets the scarcity of labeled training data by automating the feedback loop and decomposing the pipeline into two phases: a *production stage* and a *selection stage*. To align the system with the task's official pairwise evaluation protocol, the second phase implements an internal Chatbot Arena inspired by the methodology proposed by Chiang et al. (2024).

During the trial phase of the competition, however, our team observed that the production stage struggled to generate a truly diverse set of jokes, despite extensive experimentation with different prompts, models, and hyperparameter configurations. This observation led us to hypothesize that, given the intrinsic complexity of humor generation, prompt-based approaches may produce outputs with limited variability even when model architectures, decoding parameters, and prompt formulations are modified. To empirically test this hypothesis, we deliberately deviated from the intended pipeline during the final submission phase. Rather than submitting the highest-ranked jokes that were identified during the selection stage, we instead submitted those that were ranked the lowest by the arena-based evaluation. The results of the official evaluation campaign supported our hypothesis. Our system tied for Rank 1 with an overall rating of 1060, indicating that the selected outputs performed on par with those of other leading submissions despite being intentionally chosen as the weakest candidates within our internal rank-

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 3156–3164

ing. Further comparative analysis of the generated candidates revealed a high degree of lexical similarity among jokes, reinforcing the observation that LLMs still face significant challenges when dealing with tasks that require computational creativity.

2 Related Work

Humor generation remains a challenge for AI due to the requisite creativity and timing. Jentzsch and Kersting (2023) note that models like GPT often produce formal, unfunny outputs, while Horvitz et al. (2024) find that generating humor is significantly harder than removing it. To address single-pass limitations, recent work suggests specialized prompting to foster "leaps of thought" (Zhong et al., 2024), and architectural adjustments like two-phase generation (Franceschelli and Musolesi, 2024) or temperature variation (Evstafev, 2025) to maximize candidate joke potential.

Concurrently, the "LLM-as-a-judge" (Zheng et al., 2023) paradigm is increasingly adopted for evaluation. Approaches range from identifying punchlines in stand-up (Romanowski et al., 2025) to explaining multimodal humor (Hwang et al., 2025). While ChatBot Arena (Chiang et al., 2024) highlights the value of human preferences in creative tasks, HumorBench (Narad et al., 2025) validates that advanced reasoning models can reliably evaluate implicit humorous connections, serving as a scalable alternative to human assessment.

3 INF-rsrs

3.1 Task Description

The SemEval 2026 Task 1 focuses on computational humor generation, a challenging and under-explored field of natural language generation. The task is divided into two subtasks: the first is exclusively text-based, whereas the second involves multimodal generation, incorporating both textual and visual elements.

Subtask 1 required generating jokes from two types of input: a headline or a pair of words across three languages: English, Spanish, and Chinese. For headlines, the generated joke must be clearly related to the provided text (*i.e.,*, function as a punchline or as an inspired extension). For word-pair inputs, both words must be explicitly incorporated into the joke for it to be considered valid. This paper specifically addresses this subtask in English.

3.2 Solution Overview

INF-rsrs pipeline was structured in a two-phase process comprising a production and a selection stage, as depicted in Figure 1. The production stage aims to generate a diverse set of candidate jokes to prove that our theorized ceiling is model-independent, given a competent enough model. To this end, a diverse set of model configurations is employed, spanning multiple model families and varying hyperparameter settings. The selection stage was designed to identify the worst joke generated in the production stage, so we can perform an analysis to prove if the joke is still comparable to the rest. An LLM-based agent evaluates by voting in a Chatbot Arena–style (Chiang et al., 2024) competition, through which the lowest-scoring joke is selected.

3.2.1 Production Stage

Jokes are generated from an LLM prompted by the role of a comedian. The comedian prompt, presented in Appendix A.1, is structured as a directive describing the main task, followed by instructions specific to each input type (headline or word pairs), and a set of constraints to be observed by the LLM. These constraints include a restriction to the maximum number of sentences in the joke, the absence of explanations or additional text in the response, an emphasis on wit and cleverness, and the requirement that the input play an integral role in the joke.

For headline inputs, the model is instructed to generate a joke related to the headline while preserving its context, either by incorporating a punchline or by using the input as inspiration. An excerpt of this prompt is provided in Appendix A.2. For word pair inputs, the model is required to generate a joke that explicitly incorporates both words, employing one of the 11 comedy techniques of Scott Dikkers' Funny Filters (Dikkers, 2015), which is deemed the most appropriate for the given pair. An excerpt of this prompt is presented in Appendix A.3.

3.2.2 Selection Stage

The main objective of the selection stage is to rank the candidate jokes produced during the production stage for each input task, ranking them from best to worst, thereby facilitating the identification of a candidate, which in our case is the worst joke, that is subsequently selected as the final output. An overview of this stage is depicted in Figure 2. The ranking was produced by a judge agent that evaluated all jokes associated with each input through

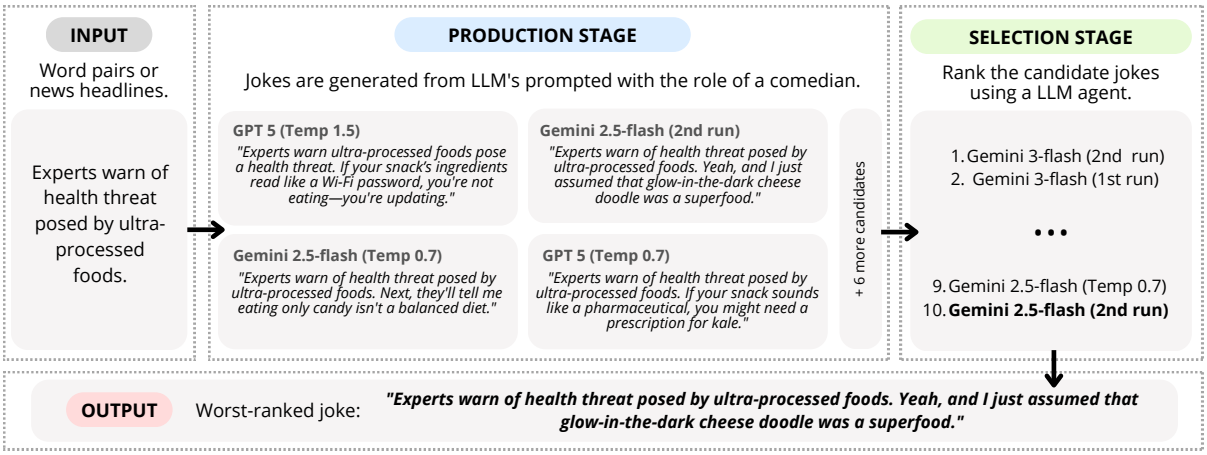

Figure 1: Overview of INF-rsrs, with each production stage representing the generation of a single candidate joke, using a combination of Model and Hyperparameters. Which are then ranked in the selection stage, ending with a singular worst joke.

pairwise (1-on-1) comparisons, following a Chatbot Arena–style competition framework (Chiang et al., 2024). For each comparison, the judge determined which of the two candidates constituted the superior joke within the given context. After n matches were conducted among each set of candidate jokes, the outcomes of all pairwise matches were subsequently aggregated to compute a final score for each candidate. At the end of this phase, we have successfully created a reliable rank for each of the inputs, from which we select the worst-ranked joke and can proceed to perform our full analysis to verify the authenticity of our claims.

The joke pairs for each match were chosen according to the active sampling rule proposed by Chiang et al. (2024), by which a pair is chosen proportionally to the expected reduction in the size of the confidence interval resulting from its evaluation. This approach yields faster convergence of the ranking procedure while retaining statistical validity, as demonstrated in the analysis performed by the original authors.

For the rankings, the BT score was used to evaluate each candidate's performance relative to the others. The BT score is based on the Bradley-Terry coefficients (Bradley and Terry, 1952) and was proposed by Chiang et al. (2024) with a design intended for arena-like scenarios. Furthermore, since both the sampling and ranking methods are derived from Chatbot Arena's work, they are integrated, providing an efficient, reproducible architecture for generating reliable ranks in a small number of matches.

The judge prompt follows a structure analogous to that of the comedian prompt. The complete prompt designed for this agent is presented in Appendix B.1. The LLM is tasked with assessing which of the two candidate jokes is better within the given context. Each candidate is evaluated with respect to its compliance with the input and its overall quality, according to three general criteria: humor, cleverness, and conciseness.

To assess compliance, specific evaluation criteria are defined. For headline inputs (Appendix B.2), the model evaluates whether the joke is clearly related to the headline, functions as a comedic extension, and preserves the essence of the original input. For word pair inputs (Appendix B.3), the model assesses whether both words are present, whether they are used meaningfully, and whether the connection between them is established cleverly. Based on these criteria, the model must select the best joke from the two sampled candidates.

4 Experimental Setup

4.1 Joke Generation

A total of 10 comedian agent configurations across three model families were evaluated. The base models used are Gemini 2.5 Flash, Gemini 3 Flash, and GPT-5, under varying temperature settings. For each configuration, the comedian agent is instantiated with a fixed model and sampling temperature.

To select this set of models, a trial phase was conducted in which a broader set was evaluated, including GPT-4o, Gemma 3, and Claude Sonnet 4.5, alongside the models ultimately selected. Based on the team's qualitative assessment of approximately 30 jokes generated per model, the models judged

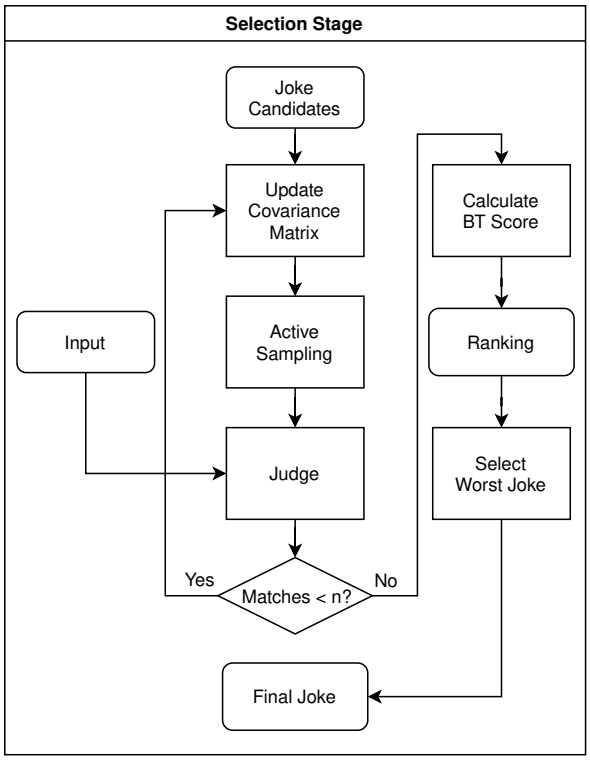

Figure 2: Overview of INF-rsrs's Selection Stage. The candidates are sampled using active sampling, with the selected match being evaluated by the judge model. This process repeats for n matches, then the results are used to calculate the BT Scores, producing a final ranking, from which the worst joke is selected as the final joke.

to produce the best joke quality were selected to be used in the final joke generation phase.

For Gemini 2.5 Flash and GPT-5, two sampling temperatures are included $T \in \{0.7, 1.5\}$ to examine the effect of output stochasticity on joke quality. A temperature of $T = 1.5$ was chosen to encourage more creative and unexpected outputs, as higher temperatures increase the diversity of the model's sampling distribution. A temperature of $T = 0.7$, on the other hand, represents a more conservative setting that keeps the output closer to the model's most probable predictions. Furthermore, for each model's default temperature ($T = 1.0$), jokes were sampled twice to increase the volume of generated data and provide a more representative sample of each model's output in its standard configuration. The complete set of configurations is summarized in Table 1.

4.2 Joke Selection

The arena-style competition consisted of $n = 25$ pairwise matches per input, conducted among $k = 10$ candidate jokes. The Bradley-Terry scores and the associated covariance matrix were recomputed after every match (*i.e.*, batch size of 1), allowing the active sampling strategy to use up-to-date uncertainty estimates at each step. No random pairing warm-up phase was employed.

Model	Temperature (T)	Run(s)
Gemini 2.5 Flash	default	2
Gemini 2.5 Flash	0.7	1
Gemini 2.5 Flash	1.5	1
Gemini 3 Flash	default	2
Gpt-5	default	2
Gpt-5	0.7	1
Gpt-5	1.5	1

Table 1: Model configurations used for joke generation. Additional runs at $T = 0.7$ and $T = 1.5$ were conducted for Gemini 2.5 Flash and GPT-5 to examine the effect of sampling temperature on joke quality.

In this competition phase, a judge agent determined which of two jokes (in a 1-on-1 competition) was better. To this end, a new model, the DeepSeek V3, was introduced as the judge. This aims to reduce a potential bias by having a model evaluate jokes generated by itself. We do not claim that such bias exists, as no controlled tests were conducted to assess this fact. However, to avoid favoring any particular model configuration, this policy was adopted. The temperature used for the judge was also set to $T = 0.2$ to provide a more deterministic response to the input jokes.

5 Results

Table 2 presents the official results of SemEval 2026 – Task 1, Subtask 1. In the overall task, INF-rsrs was ranked fifth out of 32 participants, by its raw rating. However, due to the overlap of the 95% confidence interval, our system officially shares the first-place position with seven other top-performing competitors. This was possible even by selecting the worst joke that our model configurations generated, indicating that even though there might be a better configuration, the generation of humor is still a very stale task, and LLMs can not handle major creative tasks, such as humor. Another factor to consider is the number of competitors who shared first place, suggesting that a possible ceiling may have been reached in the direct use of such models for creativity tasks. Although we do not claim that LLMs are not the way forward when it comes to computational humor generation, we strongly

Rank	System	Rating	95% CI	Votes
1	baseline	1081	[1045, 1110]	382
1	Competitor1	1080	[1046, 1120]	388
1	Competitor2	1079	[1057, 1115]	374
1	Competitor3	1063	[1036, 1099]	382
1	**INF-rsrs**	**1060**	**[1027, 1091]**	**376**
1	Competitor4	1045	[1018, 1073]	382
1	Competitor5	1041	[1009, 1064]	389
1	Competitor6	1041	[1008, 1068]	382
1	Competitor7	1034	[1005, 1072]	389
2	Competitor8	1029	[1001, 1053]	408
⋮	⋮	⋮	⋮	⋮
			Total Votes:	6,239

Table 2: Task leaderboard with the top 10 systems.

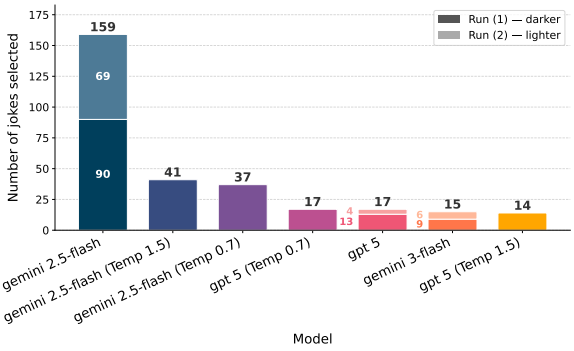

Figure 3: Histogram of selected jokes per model configuration. With configurations that had two candidate jokes, having their bar split into two for each of their executions.

believe that, even with newer and more powerful general-purpose models, there is a limit that one can achieve through the means of prompting. And we believe the results of the current task portray this limit.

To understand the dynamics of the selection stage, we analyzed the most frequently chosen model–temperature configurations. Figure 3 shows of this analysis. The examination of the seven configurations indicates that Gemini 2.5 Flash (using $T = 1.0$) yielded the highest frequency of jokes selected during this stage. Although this result may suggest inferior performance, it is important to note that this configuration generated two candidate jokes per execution, whereas configurations with alternative temperature settings produced only one. Despite this asymmetry, Gemini 2.5 Flash still consistently remained the worst-performing model across different parameter settings.

As is typical in arena-based evaluations, reliable rankings require an extensive sample of matches. Given the constraint of 25 matches per input, and despite strategies implemented to accelerate convergence, these results do not constitute a definitive proof of superiority. However, the data show that jokes generated by Gemini 2.5 were consistently ranked as the worst.

When analyzing the performance of each individual model configuration, one may conclude that Gemini 2.5 is the worst model, as it accounted for 79% of the worst jokes selected. We claim, nevertheless, that there is no such worst model and that jokes generated were of similar quality, even when generated by less powerful, and supposedly worse, models. For this, we conducted a comparison between the worst joke and each of the others using one of the most renowned metrics of NLP, the BERTScore, which is a metric used to evaluate the similarity between two sentences. The worst joke of each input was compared with each of the others, with the mean of all scores being taken into account. In a total of 300 inputs, with 10 jokes per input, we arrived at a mean score of 0.79, a standard deviation of 0.04, and 83% of pair comparisons having a BERTScore superior to 0.75.

The heatmap in Appendix C provides a per-instance view of the BERTScore similarity between the worst selected joke and each of the remaining nine model-generated jokes. Across all 300 instances and 2,700 evaluated pairs, the color distribution is remarkably uniform, with the vast majority of cells falling within a narrow high-similarity band (BERTScore $\in [0.75, 0.95]$). Crucially, no systematic pattern of low-scoring columns emerges: there is no single model that consistently produces jokes semantically distant from the worst joke, nor any particular instance where all models fail simultaneously. This global uniformity supports the hypothesis that the generation process, across all models and prompting conditions, produces jokes that are semantically close to one another.

6 Conclusion

In this paper, we present our view on the current landscape and the limitations of LLMs when used for creative purposes, such as humor generation. Following our analysis after the evaluation phase of the task, we believe that the use of LLMs as a direct means of humor generation presents a hard ceiling that is inescapable through the use of simple prompting. Therefore, we strongly believe that, although it might have slightly improved our placement, the use of the best generated jokes, in place of the worst ones, would have no significant impact

to catapult it to a solo first place winner, as per the aforementioned creative ceiling.

Future work could further explore our claims: evaluating other LLMs and exploring a broad range of prompts. Furthermore, we would like to improve the generation phase by incorporating different types of techniques such as reinforcement learning, controllable text generation, and hybrid human–AI co-creation frameworks, striving to escape the LLM generation ceiling.

Acknowledgments

This work has been partially funded by CNPq-Brazil and Capes Finance Code 001.

References

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Karyn Buxman. 2008. Humor in the or: A stitch in time? *AORN journal*, 88(1):67–77.

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)*.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: an open platform for evaluating llms by human preference. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

S. Dikkers. 2015. *How to Write Funny: Your Serious Step-by-step Blueprint for Creating Incrediby, Irresistibly, Successfully Hilarious Writing*. How to Write Funny. Scott Dikkers.

Evgenii Evstafev. 2025. Optimizing humor generation in large language models: Temperature configurations and architectural trade-offs. *arXiv preprint arXiv:2504.02858*.

Giorgio Franceschelli and Mirco Musolesi. 2024. Creative beam search: Llm-as-a-judge for improving response generation. *arXiv preprint arXiv:2405.00099*.

Jack Hessel, Ana Marasović, Jena D Hwang, Lillian Lee, Jeff Da, Rowan Zellers, Robert Mankoff, and Yejin Choi. 2023. Do androids laugh at electric sheep? humor "understanding" benchmarks from the new yorker caption contest. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 688–714.

Zachary Horvitz, Jingru Chen, Rahul Aditya, Harshvardhan Srivastava, Robert West, Zhou Yu, and Kathleen McKeown. 2024. Getting serious about humor: Crafting humor datasets with unfunny large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 855–869.

EunJeong Hwang, Peter West, and Vered Shwartz. 2025. Bottlehumor: Self-informed humor explanation using the information bottleneck principle. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22611–22632.

Sophie Jentzsch and Kristian Kersting. 2023. Chatgpt is fun, but it is not funny! humor is still challenging large language models. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 325–340.

Reuben Narad, Siddharth Suresh, Jiayi Chen, Pine SL Dysart-Bricken, Bob Mankoff, Robert Nowak, Jifan Zhang, and Lalit Jain. 2025. Which llms get the joke? probing non-stem reasoning abilities with humorbench. *arXiv preprint arXiv:2507.21476*.

Adrianna Romanowski, Pedro HV Valois, and Kazuhiro Fukui. 2025. From punchlines to predictions: A metric to assess llm performance in identifying humor in stand-up comedy. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 36–46.

Ritsu Sakabe, Hwichan Kim, Tosho Hirasawa, and Mamoru Komachi. 2025. Assessing the capabilities of llms in humor: A multi-dimensional analysis of oogiri generation and evaluation. *arXiv preprint arXiv:2511.09133*.

Brandon M Savage, Heidi L Lujan, Raghavendar R Thipparthi, and Stephen E DiCarlo. 2017. Humor, laughter, learning, and health! a brief review. *Advances in physiology education*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Shanshan Zhong, Zhongzhan Huang, Shanghua Gao, Wushao Wen, Liang Lin, Marinka Zitnik, and Pan Zhou. 2024. Let's think outside the box: Exploring leap-of-thought in large language models with creative humor generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13246–13257.

A Prompts used in Production Stage

A.1 Comedian default prompt

```
You are a masterful comedian who creates clever, concise jokes.

TASK: Create the best joke given a human input. Your joke will be judged and evaluated. If it isn't
    deemed worthy, you'll have another chance to improve it.

[SPECIFICATIONS FOR EACH INPUT TYPE]

CONSTRAINTS:
- Keep your joke SHORT: maximum 3 sentences.
- The human input must play an integral role in the joke.
- Focus on wit and cleverness over length.
- Avoid explaining the joke or adding commentary.
- Only respond with your joke, nothing else.
```

A.2 Headline specification

```
The user will input a headline. Your joke must be related to the given headline (it could be a
    punchline, or a joke inspired by it).
IMPORTANT: Preserve the context, theme, and subject matter of the headline. The joke should feel
    like a natural comedic extension of the original headline's topic.
```

A.3 Word Pairs specification

```
The user will input two words. Your joke MUST include BOTH words explicitly in the text.

You can use one of these comedy techniques (Scott Dikkers' Funny Filters):
1. Irony - Say the opposite of what you mean, or show an unexpected contradiction.
2. Character - Give personality or human traits to objects/situations.
3. Shock - Use surprising or absurd twists that catch the reader off guard.
4. Parody - Mock or exaggerate something familiar (a genre, trope, or cliché).
5. Hyperbole - Exaggerate wildly for comedic effect.
6. Wordplay (Pun) - Exploit double meanings or similar-sounding words.
7. Analogy - Compare two unrelated things in an absurd or clever way.
8. Madcap (Absurdity) - Embrace pure nonsense and surreal scenarios.
9. Misplaced Focus - Focus on the wrong detail or miss the obvious point.
10. Reference - Allude to pop culture, famous quotes, or well-known situations.
11. Paradox - Present a self-contradicting or logically impossible situation.

Choose the technique that works best for the given word pair.
```

B Prompts Selection Phase

B.1 Judge default prompt

```
You are an expert joke evaluator. Your job is to select the better joke from 2 candidates based on
    specific criteria.

[SPECIFICATIONS FOR EACH INPUT TYPE]

GENERAL QUALITY CRITERIA:
- Humor: Is it actually funny? Does it have a clear comedic payoff?
- Cleverness: Is the joke witty or just obvious?
- Conciseness: Is it tight and well-crafted, or rambling?

You will receive the ORIGINAL INPUT and two jokes (JOKE1 and JOKE2) to select which is the better
    one.
You MUST answer with ONLY a single number 1 or 2, indicating which joke was the best in the given
    context.
Do not include any explanation, context, or additional text - just the number.
```

B.2 Headline specification

```
HEADLINE CRITERIA - The joke MUST:
- Be clearly related to the headline's topic and context
- Work as a comedic extension of the headline (punchline, commentary, or inspired twist)
- Preserve the essence/theme of the original headline
```

B.3 Word Pairs specification

```
WORD PAIR CRITERIA - The joke MUST:
- Include BOTH given words explicitly in the text (automatic 0 if missing)
- Use the words in a meaningful way, not just shoehorned in
- Create a clever connection between the two words
```

C BERTScore per Instance and Model

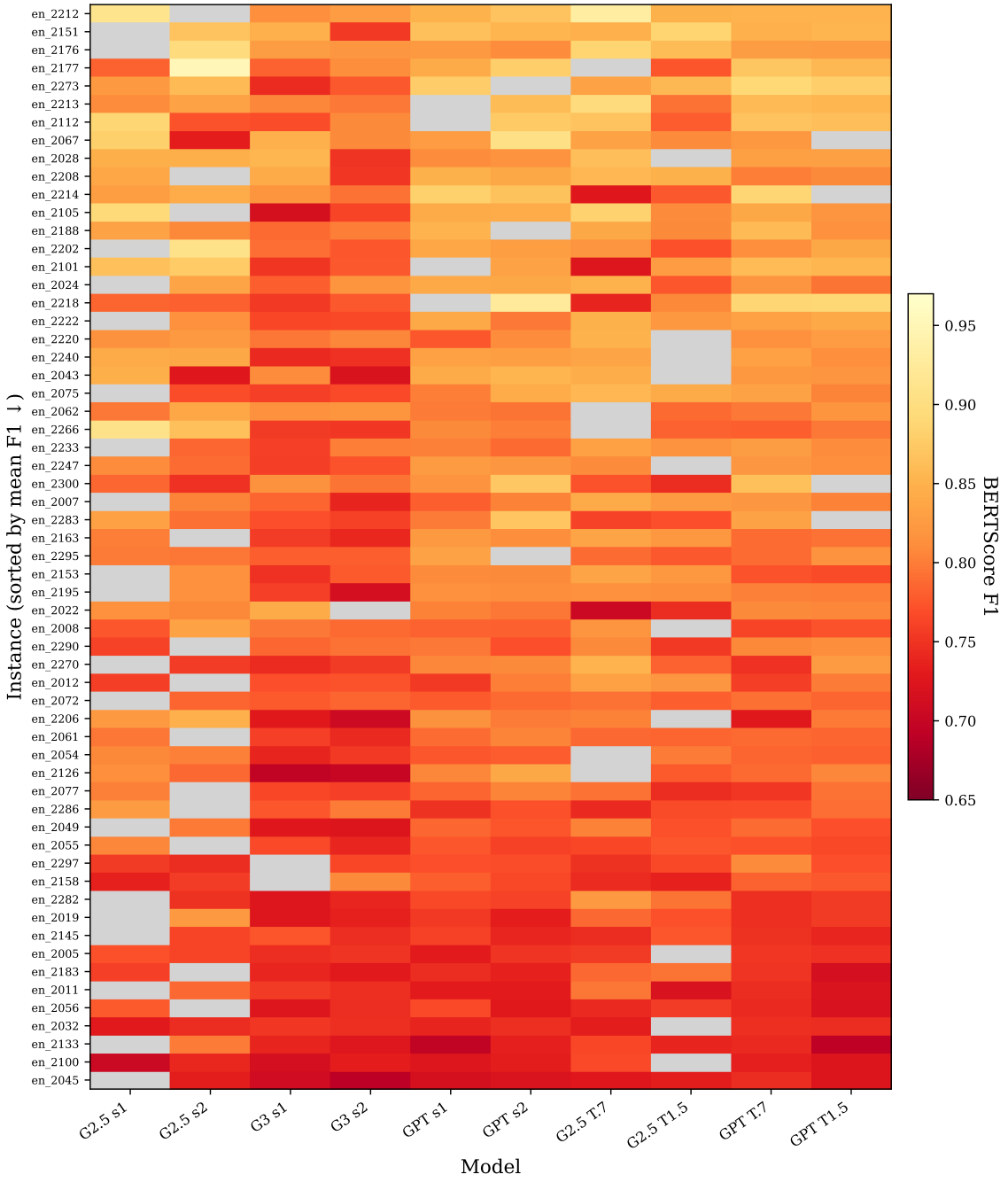

Figure 4: BERTScore F1 heatmap comparing the worst joke against the nine remaining model-generated jokes for each of the 300 instances. Rows correspond to instances sorted by descending mean F1 and columns correspond to models. Grey cells mark the model that produced the selected worst joke, excluded from computation. Color intensity reflects BERTScore F1 (yellow = high similarity, red = low similarity). The narrow color range and absence of systematic low-scoring patterns indicate that generated jokes are semantically similar regardless of model or quality ranking.