JCT at SemEval-2026 Task 1: Let the Best Joke Win - A Generate-and-Rank Approach to Constrained Humor

Batya Schechter Sarah Barzel Liebeskind Chaya
Department of Computer Science, Jerusalem College of Technology
21 Havaad Haleumi St., 91160
Jerusalem, Israel
batyabarzel@gmail.com, sarahb@shulman.org.il, liebchaya@gmail.com

Abstract

We present a humor generation system for SemEval-2026 Task 1, Subtask A (Castro et al., 2026) that produces short jokes under lexical or headline-based constraints. For each input, our system generates multiple candidate jokes using a large language model across diverse humor styles and prompting strategies, including zero-shot, few-shot, and structured prompting. Constraint satisfaction is explicitly enforced, either by requiring exact lexical inclusion or by approximating semantic relevance to a headline using sentence-embedding similarity. All valid candidates are ranked using a weighted humor score that combines semantic incongruity, emotion-based humor potential, irony likelihood, linguistic fluency, and novelty with respect to a large external jokes corpus, and the single highest-scoring joke is selected for each constraint. This approach follows a best-candidate selection paradigm, leveraging automated humor proxies to improve joke quality without task-specific fine-tuning.

1 Introduction

Humor generation remains a challenging problem in natural language generation (NLG), as it requires not only linguistic fluency but also subtle semantic incongruity, contextual awareness, and stylistic variation. Unlike many NLG tasks, humor is inherently subjective and culturally dependent, making both generation and evaluation particularly difficult for automated systems.

In this work, we focus exclusively on the English setting of the shared task

The SemEval-2026 Task 1, Subtask A addresses humor generation under explicit constraints, requiring systems to produce short jokes that either include predefined lexical items or remain semantically related to a given headline. These constraints significantly increase task difficulty, as they restrict the model's expressive space while still demanding a coherent and humorous outcome.

Recent advances in large language models (LLMs) have substantially improved the quality of open-ended text generation, including creative tasks such as joke writing. However, LLM outputs remain highly variable: while some generations are genuinely humorous, many others are bland, incoherent, or only weakly amusing. This variability is further amplified under hard constraints, where naive prompting often fails to reliably produce high-quality humor.

In this work, we propose a generate-and-select approach for constrained humor generation. Instead of relying on a single prompt or generation, our system produces multiple candidate jokes for each constraint using diverse prompting strategies and humor styles. We then automatically rank all valid candidates using a set of humor-oriented proxy metrics and select the single highest-scoring joke for submission. This strategy allows us to leverage the creative potential of LLMs while mitigating their inconsistency, without requiring task-specific fine-tuning.

2 Related Work

Computational humor has long been recognized as a challenging subfield of natural language processing (Jentzsch and Kersting, 2023). Early approaches focused on rule-based systems and linguistic templates, often targeting specific joke types such as puns or riddles (Amin and Burghardt, 2020). While these methods offered interpretability, they lacked scalability and stylistic diversity (Liu et al., 2021).

With the rise of neural language models, data-driven approaches to humor generation have become increasingly common (Amin and Burghardt, 2020; Hossain et al., 2020). Sequence-to-sequence and transformer-based models have been applied to joke generation, often trained on large corpora of humorous text (Miller et al., 2017). However,

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 89–94

such models typically struggle with controllability, particularly when required to satisfy explicit constraints (Schall and De Melo, 2025).

More recently, LLMs have demonstrated impressive zero-shot and few-shot capabilities for creative text generation (Jentzsch and Kersting, 2023). Several works have explored prompting strategies for humor generation, style control, and constrained generation (Jentzsch and Kersting, 2023; Schall and De Melo, 2025). Nevertheless, prior work has shown that LLM outputs are highly sensitive to prompt formulation and often require extensive prompt engineering to achieve consistent quality (Zhao et al., 2021).

Our work differs from prior approaches by framing constrained humor generation as a selection problem rather than a pure generation problem (Hossain et al., 2017). Instead of attempting to generate a single optimal joke, we generate multiple diverse candidates and rely on automated humor proxies to select the best-performing output for each constraint.

3 Methodology

3.1 Task Overview

Each input instance consists of an identifier and either (1) two lexical items that must appear in the generated joke, or (2) a headline to which the joke should be semantically related. The output is a short joke of one to three sentences.

3.2 Candidate Generation

For each input constraint, we generate multiple candidate jokes using a large language model provided by OpenAI - gpt-5.1[1]. To encourage diversity, we vary both the prompting strategy and the humor style. Specifically, we experiment with: (i) zero-shot prompting, where the model is instructed to produce a short joke in a specified humor style; (ii) few-shot prompting, where two brief style-specific example jokes are provided to steer generation; and (iii) chain-of-thought–guided prompting, where the model is guided to plan the joke (e.g., identify the humor mechanism and select a punchline direction) before producing the final output. We use ten humor styles: Wordplay, Irony, Incongruity, Absurd, Observational, Self-deprecating, Social, Logical twist, Dark humor, and Wholesome. Each candidate is generated independently for every (strategy, style) combination, resulting in a total of 30 candidates per constraint. The exact prompt templates are provided in Appendix A.

3.3 Constraint Enforcement

Generated jokes are filtered to ensure compliance with the task constraints. For lexical constraints, we require the exact inclusion of both specified words. For headline-based constraints, we approximate semantic relevance using cosine similarity between sentence embeddings of the headline and the generated joke, and we require the similarity to exceed a fixed threshold (0.6 in our experiments). Candidates that fail to meet the constraint are discarded or regenerated up to a fixed retry limit (4 times in our experiments)

4 Ranking and Selection

To estimate joke quality automatically, we compute several complementary humor-oriented proxy metrics using pretrained neural models.

Semantic incongruity. We estimate incongruity as the inverse cosine similarity between the semantic representations of the joke's setup and punchline, computed using a Sentence-BERT model[2]

Humor potential. We use a pretrained emotion classification model RoBERTa fine-tuned on GoEmotions[3] to estimate humor-related affect, summing the predicted probabilities of amusement- and surprise-related emotions.

Irony likelihood. We apply a pretrained irony detection model based on RoBERTa [4] to estimate the presence of ironic or sarcastic cues in the generated joke.

Fluency and readability. Linguistic fluency is measured via the negative perplexity under a pretrained GPT-2 language model[5].

Novelty. We compute semantic similarity between the generated joke and a large external corpus of jokes[6] using sentence embeddings from a

[1] https://developers.openai.com/api/docs/models/gpt-5.1

[2] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[3] https://huggingface.co/SamLowe/roberta-base-go_emotions

[4] https://huggingface.co/cardiffnlp/twitter-roberta-base-irony

[5] https://huggingface.co/openai-community/gpt2

[6] https://www.kaggle.com/datasets/abhinavmoudgil95/short-jokes

Sentence-BERT model[7], defining novelty as the inverse of the mean similarity to the most similar examples.

4.1 Weighted Scoring and Selection

All metric scores are first normalized to a common scale and then combined into a single weighted aggregate score. For a candidate joke c, we define:

$$S(c) = \sum_{m \in \mathcal{M}} w_m \, \tilde{s}_m(c) \tag{1}$$

where $\tilde{s}_m(c)$ denotes the normalized score of metric m for candidate c, and $\mathcal{M} = \{\text{inc}, \text{hum}, \text{nov}, \text{flu}, \text{iro}\}$ represents semantic incongruity, humor potential, novelty, fluency, and irony likelihood, respectively. We use fixed weights $w_{\text{inc}} = 0.35$, $w_{\text{hum}} = 0.25$, $w_{\text{nov}} = 0.15$, $w_{\text{flu}} = 0.15$, and $w_{\text{iro}} = 0.10$.

For each input constraint, we generate a candidate set $\mathcal{C}$ and select the final output by:

$$c^* = \arg\max_{c \in \mathcal{C}} S(c). \tag{2}$$

We selected the weights manually based on qualitative inspection of outputs during system development, using a small set of development examples from the trial phase. Our goal was to prioritize metrics that most consistently aligned with our intuitive judgments of joke quality, while still preserving fluency and novelty. In particular, we assigned the highest weight to semantic incongruity, as it appeared most closely related to the presence of a punchline or humorous twist, followed by humor potential. We did not perform a large-scale supervised tuning procedure due to the absence of labeled funniness judgments.

This best-candidate selection paradigm exploits diversity at generation time while enforcing quality at selection time.

5 Experiments and Analysis

We analyze the selected outputs of our generate-and-select framework by examining the distribution of winning jokes across generation methods and humor styles. For each input constraint, a single joke is chosen based on the highest weighted humor score.

Figure 1 shows the distribution of selected jokes by generation method. Few-shot prompting is selected most frequently, followed by chain-of-thought–guided prompting, while zero-shot prompting is chosen substantially less often. This indicates that even limited guidance—either via style-specific examples or structured prompting—improves the reliability of humor generation under constraints, compared to unguided zero-shot generation.

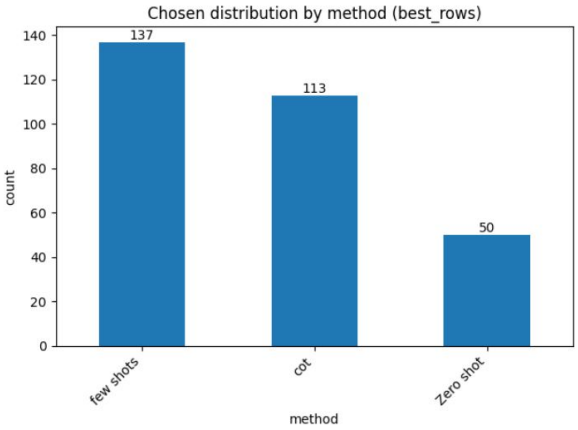

Figure 1: distribution by method

Figure 2 presents the distribution of selected jokes by humor style. Observational humor dominates the selected outputs, with irony and self-deprecating humor also appearing frequently. In contrast, more abstract styles such as incongruity, wordplay, and wholesome humor are selected less often. This pattern suggests that humor styles grounded in everyday situations and clear pragmatic cues are more consistently favored by the automatic ranking metrics, whereas subtler or structurally complex forms of humor are harder to capture reliably.

Overall, the results indicate that both prompting strategy and humor style significantly influence the final selection, reflecting the interaction between generation diversity and the automated ranking mechanism rather than absolute judgments of humor quality.

6 Limitations

Our approach relies on automatic proxy metrics to estimate humor quality, which cannot fully capture the subjective, cultural, and contextual aspects of human humor. While humor datasets do exist, obtaining large-scale, reliable annotations for perceived funniness and preference in generated jokes remains difficult. This limits both supervised optimization of the ranking function and direct validation against human judgments

[7] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

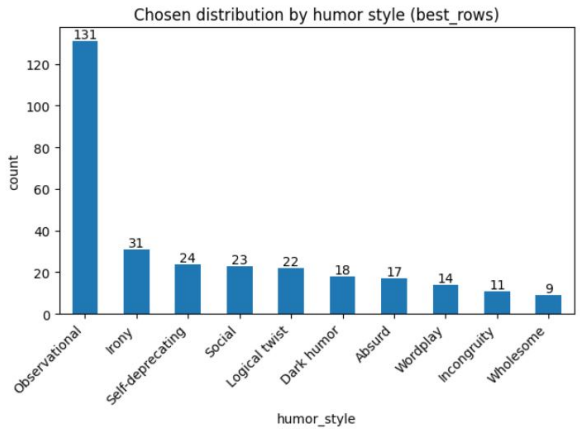

Figure 2: Distribution of selected jokes by humor style

Although the selected metrics reflect complementary properties such as incongruity, emotional response, fluency, and novelty, they may favor certain humor styles that align well with pretrained models while underestimating subtler or less conventional forms of humor. In addition, the weighting of the individual scoring components is manually defined and fixed across all constraints, due to the absence of sufficient labeled data for systematic tuning and performance-based optimization.

Finally, our evaluation does not include human judgments, as we do not have access to a large and diverse pool of annotated examples. This further limits our ability to directly assess perceived funniness or audience preference and to compare alternative weighting schemes in a controlled manner. Incorporating human evaluation and data-driven or learned weighting strategies remains an important direction for future work.

7 Conclusion

We introduced a generate-and-select framework for constrained humor generation that leverages large language models and automated humor proxies to select a single high-quality joke per constraint. Our results suggest that ranking diverse candidates is an effective strategy for mitigating LLM variability in creative tasks.

Notably, our system was ranked tied for first place as one of eight top-performing systems in the organizers' rating-based evaluation at SemEval-2026, highlighting the strong practical effectiveness of our approach in a real-world evaluation setting.

Future work may explore adaptive weighting, human-in-the-loop selection, and cross-lingual humor generation.

8 Code Availability

The source code for our system is available at: https://github.com/batyaSchechter/Humor_generation

9 Appendices

A Prompting Details

This appendix presents the exact prompt templates used for joke generation in our experiments. All prompts were instantiated by filling in the placeholders with the corresponding humor style and constraint values.

A.1 Zero-Shot Prompt

```
You are a humor writer. Write one short,
    funny joke, {humor_style} style,
that naturally includes the words "{word1}"
    and "{word2}".
The joke should be coherent, sound natural,
    and make sense in context.
It must include both words exactly as given.
(1-3 sentences). Make sure it's a complete
    joke with a punchline.
Output only the joke.
```

For headline-based constraints, the lexical condition is replaced with:

```
You are a humor writer. Write one short,
    funny joke, {humor_style} style,
that is related to the headline:
    "{headline}".
The joke should be coherent, sound natural,
    and make sense in context.
(1-3 sentences). Make sure it's a complete
    joke with a punchline.
Output only the joke.
```

A.2 Few-Shot Prompt (Two Examples)

```
You are a humor writer. Write one short,
    funny joke, {humor_style} style,
that naturally includes the words "{word1}"
    and "{word2}".
The joke should be coherent, sound natural,
    and make sense in context.
It must include both words exactly as given.
(1-3 sentences). Make sure it's a complete
    joke with a punchline.
Example {humor_style} jokes:
1: {example_1}
2: {example_2}
Output only the joke.
```

For headline-based constraints:

```
You are a humor writer. Write one short,
    funny joke, {humor_style} style,
```

```
that is related to the headline:
    "{headline}".
The joke should be coherent, sound natural,
    and make sense in context.
(1-3 sentences). Make sure it's a complete
    joke with a punchline.
Example {humor_style} jokes:
1: {example_1}
2: {example_2}
Output only the joke.
```

A.3 Structured (Chain-of-Thought–Guided) Prompt

```
Write a joke that naturally includes the
    words "{word1}" and "{word2}"
in the humor style: {humor_style}.

First, use chain-of-thought reasoning:
1. Identify what makes this humor style
    funny.
2. Describe the structure of a typical joke
    in this style.
3. Brainstorm 3 possible directions for the
    joke.
4. Choose the strongest idea and explain
    why.

Then, write the final joke in one short
    sentence.
Do NOT include the chain-of-thought
    reasoning in the final answer.
```

For headline-based constraints:

```
Write a joke that is related to the
    headline "{headline}"
in the humor style: {humor_style}.

First, use chain-of-thought reasoning:
1. Identify what makes this humor style
    funny.
2. Describe the structure of a typical joke
    in this style.
3. Brainstorm 3 possible directions for the
    joke.
4. Choose the strongest idea and explain
    why.

Then, write the final joke in one short
    sentence.
Do NOT include the chain-of-thought
    reasoning in the final answer.
```

A.4 Humor Styles

The `{humor_style}` placeholder was instantiated with one of the following values: *Wordplay, Irony, Incongruity, Absurd, Observational, Self-deprecating, Social, Logical twist, Dark humor, Wholesome*.

A.5 Few-Shot Prompt Examples

This appendix lists the example jokes used for few-shot prompting. For each humor style, two short jokes were provided as fixed conditioning examples. All examples were drawn from publicly available joke sources.

Wordplay

- I wondered why the baseball was getting bigger. Then it hit me.
- I'm reading a book on anti-gravity. It's impossible to put down.

Irony

- I love deadlines. I especially like the whooshing sound they make as they fly by.
- I'm on a seafood diet. I see food... and I eat it.

Incongruity

- I tried to take a selfie with my fridge, but it didn't say cheese.
- My dog chased a squirrel up a tree. Too bad it was a plastic tree in the mall.

Absurd

- I told my toothbrush we were breaking up. It couldn't handle the plaque in our relationship.
- A duck walked into a pharmacy and bought lip balm. "Put it on my bill," it said.

Observational

- Isn't it weird how we talk to babies in a higher voice... even when the baby is a dog?
- You never realize how many people you dislike until you have to name your Wi-Fi.

* By observational humor, we refer to jokes grounded in familiar everyday situations, habits, or social routines, where the humor emerges from noticing common but often overlooked aspects of ordinary life

Self-deprecating

- I don't need a hairdresser. My pillow gives me a new hairstyle every morning.
- My fitness coach told me to "touch my toes." I said, "If they wanted to be touched, they wouldn't be so far away."

Social

- We now have smart TVs, smart cars, smart fridges... I miss when only people were dumb.
- Group projects taught me one thing: some people were born to ride, not to drive.

Logical Twist

- I told my friend I broke my arm in two places. He said, "Stop going to those places."
- The doctor told me I needed more exercise, so I started sleeping on the other side of the bed for variety.

Dark Humor

- My therapist says I have pre-traumatic stress disorder. I'm stressed about things that will probably happen.
- I told my skeleton friend he was too thin. He took it personally.

Wholesome

- I asked my grandma for a bookmark. She smiled and said, "Honey, I've been calling you that for years."
- My dog loves when I read aloud. I think he likes that I finally speak his language: "woof woof."

References

Miriam Amin and Manuel Burghardt. 2020. A survey on approaches to computational humor generation. In *Proceedings of LaTeCH-CLfL 2020*, pages 29–41, Barcelona, Spain (Online). Association for Computational Linguistics.

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)*.

Nabil Hossain, John Krumm, Michael Gamon, and Henry Kautz. 2020. Semeval-2020 task 7: Assessing humor in edited news headlines. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 746–758, Barcelona (online). International Committee for Computational Linguistics.

Nabil Hossain, John Krumm, Lucy Vanderwende, Eric Horvitz, and Henry Kautz. 2017. Filling the blanks (hint: plural noun) for mad libs humor. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 638–647. Association for Computational Linguistics.

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

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv*.

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, Vancouver, Canada. Association for Computational Linguistics.

Maximilian Schall and Gerard De Melo. 2025. The hidden cost of structure: How constrained decoding affects language model performance. In *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing-Natural Language Processing in the Generative AI Era*, pages 1074–1084.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *arXiv*.