j10official at SemEval-2026 Task 1: Neurosymbolic Humor Generation via GTVH-Guided LLM Decomposition

Jatin Agrawal
IIIT Hyderabad
jatin.agrawal@research.iiit.ac.in

Radhika Mamidi
IIIT Hyderabad
radhika.mamidi@iiit.ac.in

Abstract

We present a neurosymbolic pipeline for computational humor generation grounded in the General Theory of Verbal Humor. The system constructs the joke in five sequential stages: context analysis, humor architecture (identifying core incongruity), delivery strategy, content writing, and pairwise judging, orchestrated through the DSPy framework. The system generates four candidate jokes per input with independent humor strategies, then selects the best through knockout tournament-style evaluation. Despite using Gemma 3 27B, a model with roughly 20× fewer total parameters than frontier systems, our approach achieves competitive results across all five subtasks of SemEval-2026 Task 1 (MWAHAHA), placing 2nd in two subtasks. We argue that these results demonstrate the viability of structured, theory-driven decomposition for solving complex tasks and that how a model reasons about humor is just as important as how large the model is.

1 Introduction

Computational humor generation remains one of the most demanding problems in natural language generation. Unlike tasks such as summarization or translation, humor requires the coordinated construction of incongruity, surprise, timing, and stylistic control. LLMs inherently struggle with this as they are trained to maximize next-token probability, whereas humor often depends on strategically violating expectations. Just as raw cognitive capacity does not make every human a comedian, strong general reasoning in LLMs does not automatically yield strong humor without explicit comedic structure.

Accordingly, humor generation should not be treated as a single decoding problem. Core objectives such as constructing incongruity, preserving coherence, calibrating tone, and timing the punchline are partially competing, and attempting to optimize them simultaneously can lead to generative interference. To address this challenge, we decompose humor generation into modular components inspired by the General Theory of Verbal Humor (GTVH) (Attardo and Raskin, 1991; Attardo, 2001). By operationalizing GTVH's six Knowledge Resources (Situation, Target, Logical Mechanism, Script Opposition, Narrative Strategy, and Language) as dedicated stages within a unified pipeline, we transform the open-ended instruction "write a funny joke" into a sequence of constrained, interpretable subtasks. This separation of structural reasoning from surface realization reduces interference and enables more controlled humor construction.

We evaluate this approach in SemEval-2026 Task 1, MWAHAHA (*Models Write Automatic Humor And Humans Annotate*) (Castro et al., 2026), which spans five subtasks across three languages and two modalities under strict novelty constraints. Using the same architecture and underlying model across all subtasks, our system ranks 2nd on Task A (Chinese) and Task B2 (GIF + prompt completion), and 4th on Task A (English) and Task B1. These results are competitive with a baseline based on prompting Gemini 2.5 Flash, a substantially larger commercial model, suggesting that explicit structural decomposition can partially mitigate scale disadvantages in creative NLG. Code and all intermediate outputs are publicly available at https://github.com/J10Official/Semeval2026_Task1.

2 Background

2.1 Task Definition

SemEval-2026 Task 1 (MWAHAHA) (Castro et al., 2026) evaluates humor generation across text-only and multimodal settings using human preference judgments with Elo-based ranking. The task consists of two categories:

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 3015–3026

Task A: Text-Based Humor. Given textual constraints, systems generate a joke in the target language: (i) news headline-inspired humor, or (ii) word-constrained humor requiring the inclusion of two specified terms. Task A is conducted in English, Spanish, and Chinese.

Task B: Multimodal Humor. Given a GIF, systems either generate a humorous caption (B1) or complete a fill-in-the-blank prompt (B2), subject to short length constraints.

2.2 Related Work

General Theory of Verbal Humor (GTVH). Our pipeline is fundamentally grounded in the GTVH (Attardo and Raskin, 1991; Attardo, 2001), which extends Raskin's Semantic Script Theory (Raskin, 1985) and characterizes jokes through six hierarchical Knowledge Resources (KRs): Situation, Target, Logical Mechanism, Script Opposition, Narrative Strategy, and Language. While previous computational humor works often leave these resources latent within a model's internal representations during end-to-end generation, we explicitly operationalize these KRs as deterministic, typed computational modules using DSPy signatures. This approach bridges theoretical linguistics and neurosymbolic NLP, ensuring every generated joke possesses a valid structural incongruity and a verifiable cognitive resolution strategy before actual text is realized.

Generative Humor in the LLM Era. Early computational humor relied on rigid templates (Attardo and Raskin, 1991; Stock and Strapparava, 2003); LLMs shifted the paradigm toward end-to-end generation but frequently default to generic patterns, struggling with the high-context "understanding" humor requires (Hessel et al., 2023). In multimodal settings, where visual understanding itself remains an open challenge for large models (Caffagni et al., 2024), grounding humor in visual cues is particularly difficult (Hessel et al., 2023). We address this by explicitly modeling the reasoning process rather than relying on latent model capabilities.

Structured Reasoning and Decomposition. Chain-of-Thought prompting (Wei et al., 2022) elicits stronger reasoning in complex tasks; humor-specific approaches like *HumorChain* (Zhang et al., 2025) and structured comedy-writing methodologies (Toplyn, 2014) further show that decomposing humor into discrete steps outperforms single-turn prompting. We extend this line by implementing decomposition not as prompts but as modular, typed signatures within DSPy (Khattab et al., 2023).

3 System Overview

3.1 Design Philosophy

Our design follows a neurosymbolic perspective. The symbolic layer is defined by a fixed module decomposition, typed interfaces, and alignment with the six GTVH knowledge resources. This layer specifies *what* reasoning steps must occur, while neural components determine *how* each step is realized linguistically.

Structure therefore precedes realization. The system separately models contextual setup, incongruity design, narrative strategy, and final joke writing. Multiple candidate jokes are generated and compared through tournament based judging to select the strongest output. Each module operates under explicit constraints, reducing cross-objective interference during generation.

By disentangling these components, the pipeline mitigates common failure modes of monolithic decoding, such as generic humor unanchored to the input, premature punchlines, and over-explicit explanations. Humor generation is thus implemented as a sequence of structured, theory-aligned decisions rather than a single generative step.

3.2 Architecture

The pipeline consists of five stages, implemented as typed DSPy Signatures, with Modules 2–4 instantiated in four parallel branches per input (see Figure 1; a complete walkthrough with intermediate outputs is provided in Appendix G). The same architecture is reused across all five subtasks; only the signature descriptions vary by task. This design isolates performance gains to architectural structure and controlled diversification rather than increased model scale or heterogeneous ensembling. Importantly, diversity arises at the level of humor mechanisms and narrative strategies rather than surface-level rewording.

Module 1: ContextEnricher (GTVH: Situation, Target). The ContextEnricher analyzes the raw input to extract factual subtext, implicit assumptions, and cultural context. It produces typed outputs: `situation` (elaboration of the input's factual reality and subtext)

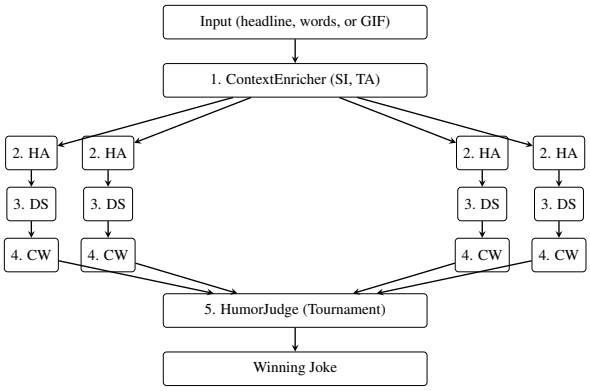


Figure 1: System architecture. Module 1 runs once per input; Modules 2–4 run in four parallel branches; Module 5 selects the winner via single-elimination tournament. HA = HumorArchitect, DS = DeliveryStrategist, CW = ContentWriter.

and `semantic_associations` (cultural references and properties linked to key terms)—that are consumed as structured inputs by all downstream modules. This module runs **once per input**, ensuring all candidate jokes are grounded in a consistent factual understanding.

Module 2: HumorArchitect (GTVH: Logical Mechanism, Script Opposition). The HumorArchitect designs the cognitive mechanism that makes the joke work without writing the actual prose. Its DSPy signature requires generating multiple typed outputs, which include `focal_targets` (pivot concepts), `cognitive_manipulation` (a specific instruction on how to execute the semantic twist), `logical_mechanism` (assigning a formal GTVH label such as False Analogy or Garden Path), `expected_script` and `opposing_script` (the normal expectation vs. the surprise reality), and the overarching `script_opposition` (the incongruity axis). This module is instantiated four times in parallel using a `ThreadPoolExecutor`, each instance producing an independent humor strategy. Variation across branches arises from independent stochastic decoding under identical prompts to guarantee diversity at the conceptual mechanism level rather than just surface rewording.

Module 3: DeliveryStrategist (GTVH: Narrative Strategy). Determines *how* to perform the humor. The Strategist consumes the Architect's logical blueprint, outputting a `strategic_analysis` (justifying why a specific format suits the generated mechanism), chooses a `narrative_strategy` (e.g., Dialogue, One-Liner, Fake News), and defines a `language_style` (e.g., Dry/Cynical, Deadpan). The incongruity logic (Module 2) and the narrative packaging (Module 3) are treated as strictly orthogonal decisions.

GTVH KR	Module	Function
Situation	ContextEnricher	Factual grounding
Target	ContextEnricher	Humor targets
Log. Mech.	HumorArchitect	Cognitive twist
Script Opp.	HumorArchitect	Incongruity
Narr. Strat.	DeliveryStrategist	Format & voice
Language	ContentWriter	Surface text

Table 1: Mapping of GTVH Knowledge Resources to pipeline modules.

Module 4: ContentWriter (GTVH: Language). Executes the joke, transforming the structural blueprint into prose. It receives all upstream outputs plus language-specific **writing guidelines**—general comedic craft principles (economy of language, setup misdirection, commitment to the bit) that are domain-general rather than tuned on task data (see Appendix C). Outputs include `draft_setup`, `draft_punchline`, and `final_joke`. The explicit draft-then-polish structure encourages the model to separate logical construction from surface polish.

Module 5: HumorJudge (Selection). The four candidates are evaluated through a **single-elimination tournament**: two parallel semi-finals, then a final (three comparisons vs. six for a full round-robin). We use pairwise comparison rather than absolute scoring because LLMs are more reliable at relative preference judgments than at assigning calibrated humor scores (Zheng et al., 2023). The judge receives language-specific **evaluation criteria** as a structured input field (see Appendix C).

Table 1 summarizes the alignment between GTVH Knowledge Resources and pipeline modules.

3.3 Multimodal Extension (Task B)

For GIF-based tasks, a preprocessing step converts GIFs into rich textual descriptions using **Nemotron Nano 12B VL** (NVIDIA, 2025), a vision-language model accessed via OpenRouter. The process is: (1) download the GIF; (2) convert GIF to MP4 via `ffmpeg`; (3) analyze the video with task-specific prompts that extract subjects, actions, expressions, and setting details (see Appendix D). No humor

reasoning is performed during preprocessing; the vision model produces purely descriptive output.

For Task B2, the fill-in-the-blank prompt is included in the vision model's instructions so that the description is contextualized. These descriptions are cached and fed into the same five-module pipeline used for text-based tasks.

3.4 DSPy Implementation

Each module is implemented as a `dspy.Signature` class with typed `InputFields` and `OutputFields`, preventing downstream modules from bypassing upstream reasoning steps (Table 2 details the comprehensive output fields mapped from the GTVH resources for each individual stage). A single `UnifiedHumorPipeline` class loads task-specific signatures from a registry and routes data through the same module sequence—no architecture change or manual tuning is needed, only system prompt specialization.

3.5 Constraint Satisfaction

Task A2 requires that both specified words appear in the generated joke. We implement a deterministic post-generation validator using flexible regex matching: case-insensitive, word-boundary-aware matching for alphabetic languages; substring matching for Chinese; and support for common morphological variants (e.g., plurals, verb forms, possessives). Candidates failing validation are excluded from the tournament. If all candidates fail, the system proceeds with judging rather than regenerating to avoid additional latency. Character limits (900 for EN/ES, 300 for ZH) and word-count constraints ($\leq$20 for Task B) are enforced deterministically.

4 Experimental Setup

We use the official MWAHAHA evaluation data without modification (Table 3). No training data is provided by design, and the system uses no labeled humor data.

The primary LLM is Gemma 3 27B IT (Gemma Team, Google DeepMind, 2025), accessed via the OpenRouter API in bf16 precision. For Task B (vision), we use Nemotron Nano 12B VL. The system is implemented in DSPy v2.4+ with LiteLLM (BerriAI, 2024) for unified API access. All modules use default decoding parameters; no hyperparameter tuning was performed. LLM caching is explicitly disabled (`litellm.cache = None`) to ensure independent stochastic generations across parallel branches.

Generation uses 4-way parallel branch sampling per item and up to 8-way parallel item processing via `ThreadPoolExecutor`. API failures are handled via a custom `@with_retry` decorator tracking specific error types (e.g., rate limits vs. transient parses), with exponential backoff (5s $\rightarrow$ 120s) and explicit `ChatAdapter` enforcement for Gemma models to accommodate JSON mode incompatibilities.

Systems are evaluated via human pairwise preference judgments with Elo-based ranking on a Chatbot Arena-style leaderboard.

5 Results

5.1 Official Rankings

Table 4 presents the official Elo scores. The shared task baseline (Gemini 2.5 Flash with simple, task-specific single-turn prompts) is provided by the organizers.

Our system achieves Elo scores within 37–125 points of the baseline, placing **2nd** on Task A (zh) ($\Delta = 37$) and Task B2 ($\Delta = 29$). Three observations contextualize these results:

- **Parameter efficiency.** Gemma 3 27B ($\sim$27B parameters) competes with Gemini 2.5 Flash ($>$200B effective parameters), suggesting that structured reasoning can partially compensate for an order-of-magnitude scale gap.
- **Architectural uniformity.** The same five-module pipeline is used for all subtasks; only the DSPy signature descriptions vary. Results therefore reflect the general effectiveness of GTVH-guided decomposition rather than task-specific engineering.
- **Cross-lingual transfer.** The 2nd-place finish on Chinese is notable given Gemma 3 27B's predominantly English training data. The explicit comedic reasoning scaffolding appears to partially compensate for weaker target-language fluency.

5.2 Analysis

We compare outputs against Gemini 3 Pro (Google, 2025) with single-turn prompts (full comparison in Tables 8–11, Appendix E). The pipeline's outputs exhibit: (1) **narrative commitment**—fully developed frames (news broadcasts, dialogues) from the DeliveryStrategist rather than generic

Module	Output Field	Type	Description
ContextEnricher	`situation`	`str`	Factual reality and subtext elaboration
ContextEnricher	`semantic_associations`	`list[str]`	Cultural references and associations
HumorArchitect	`focal_targets`	`str`	Pivot concepts for the joke
HumorArchitect	`cognitive_manipulation`	`str`	One-sentence twist instruction
HumorArchitect	`logical_mechanism`	`str`	GTVH label
HumorArchitect	`expected_script`	`str`	Normal expectation
HumorArchitect	`opposing_script`	`str`	Surprise reality
HumorArchitect	`script_opposition`	`str`	*[A] vs. [B]* format
DeliveryStrategist	`strategic_analysis`	`str`	Format compatibility analysis
DeliveryStrategist	`narrative_strategy`	`str`	Chosen delivery format
DeliveryStrategist	`language_style`	`str`	Voice/register
ContentWriter	`draft_setup`	`str`	Joke build-up
ContentWriter	`draft_punchline`	`str`	Joke reveal
ContentWriter	`final_joke`	`str`	Polished complete joke
HumorJudge	`critique`	`str`	Comparative analysis
HumorJudge	`better_joke`	`Literal`	Tournament winner ("Joke 1" or "Joke 2")

Table 2: All typed output fields per module.

Subtask	Lang.	Items	Input Type
A (en)	English	300	Headlines + word pairs
A (es)	Spanish	300	Headlines + word pairs
A (zh)	Chinese	300	Headlines + word pairs
B1	English	300	GIF URLs
B2	English	300	GIF URLs + prompts

Table 3: Official evaluation data.

Subtask	Baseline Elo	Our Elo	Rank
A (en)	1081	1005	4th
A (es)	1140	1015	6th
A (zh)	1053	1016	**2nd**
B1	1124	994	4th
B2	1022	993	**2nd**

Table 4: Official Elo scores and rankings. Baseline is Gemini 2.5 Flash (ranks 1st on all subtasks).

observations; (2) **structural clarity**—explicit `draft_setup`/`draft_punchline` separation before final composition; (3) **input specificity**—tight coupling to the input via ContextEnricher analysis; and (4) **diverse strategies**—genuine mechanism variation across the four parallel HumorArchitect branches.

Recurring failure modes include **over-elaboration** (the multi-module chain accumulates detail, particularly problematic for Chinese's 300-character limit), **cultural mismatch** (Anglo-centric strategies from English-dominant training data, likely contributing to the weaker Spanish result), and **GIF description loss** (text-mediated descriptions occasionally miss the key visual humor moment).

Mechanism diversity. Across 5,885 candidates, the HumorArchitect generates 371 unique logical mechanism labels, with Recontextualization dominant (21.6%)—consistent with GTVH's prediction that incongruity drives humor. Strategy profiles diverge by modality: text tasks favor Recontextualization, Irony, and Absurdity, while GIF captioning (B1) relies on Recontextualization and Juxtaposition and prompt completion (B2) favors Absurd Literalization and Visual Hyperbole. Task A (zh) uniquely ranks Irony first, suggesting ironic reversal transfers more reliably into Chinese conventions. Table 5 lists the top three preferred generation strategies per subtask, corroborating the presence of deliberate compositional divergence across tasks. Full distributions are reported in Appendix A.

Judge positional bias and limitations. Candidate 1 wins 47% of tournaments (expected: 25%), a known limitation of LLM-as-judge approaches (Zheng et al., 2023), which frequently suffer from strong positional and verbosity biases. The bias is most extreme for Task B1 (67%), where shorter, more similar captions may cause the judge to default to the first acceptable option. Conversely, it is most balanced for Task A (zh) (37%), suggesting that when inter-candidate linguistic diversity is higher, the model evaluates more critically. While our single-elimination tournament structure reduces the cognitive load of evaluating all candidates simultaneously (limiting it to three discrete pairwise matches), we did not employ positional swapping (A vs. B then B vs. A) during infer-

Subtask	Top-3 Mechanisms	%
A (en)	Recontextualization	17.3
	Irony	15.0
	Absurdity	14.3
A (es)	Recontextualization	26.1
	Irony	17.2
	Absurdity	13.2
A (zh)	Irony	21.7
	Recontextualization	19.9
	Absurdity	14.9
B1	Recontextualization	43.0
	Juxtaposition	21.8
	Misattribution	10.2
B2	Absurd Literalization	38.2
	Visual Hyperbole	14.4
	Mood Embodiment	12.6

Table 5: Top-3 logical mechanisms per subtask.

ence. Regardless of this primacy bias, the high correlation between the pipeline's final selections and the Elo scores awarded by human evaluators demonstrates that even a biased LLM judge successfully functions as an effective filter for quality humor. Per-subtask breakdowns are provided in Appendix B.

6 Conclusion

We presented a neurosymbolic humor generation pipeline that decomposes joke creation into five GTVH-guided modules implemented as DSPy signatures. By explicitly separating incongruity design from narrative execution, our 27B-parameter system achieves competitive results against models heavily exceeding its scale, securing 2nd place in two subtasks.

These results suggest that structured decomposition can serve as a viable alternative to model scaling for creative generation tasks. Just as a professional comedian does not simply "think of something funny" but instead identifies targets, constructs incongruities, chooses delivery formats, and revises, our pipeline makes this implicit cognitive process explicit—providing scaffolding that enables a modest-scale LLM to compete with frontier models. More broadly, our work demonstrates that the GTVH, originally a descriptive linguistic theory, can function as a prescriptive computational framework for humor generation. Rather than asking "how large must a model be to generate good humor?", we argue that the more productive question is "what cognitive structure must we provide to enable a model of any size to reason about humor effectively?"

Several directions remain open; we outline the most promising in Section 7.

7 Future Work

1. **Judge debiasing and alternative evaluations.** Relying heavily on an LLM for pairwise preference introduces significant positional bias. Future implementations should mitigate this computationally by performing bidirectional comparisons (evaluating Candidate A vs. Candidate B, then B vs. A) and randomizing presentation order. Deeper architectural solutions might involve training specialized reward models or utilizing reference-free metrics to reduce off-the-shelf LLM reliance.
2. **Module-level ablation studies.** While the pipeline achieves competitive Elo rankings, we have not rigorously quantified the individual contribution of each GTVH stage. Future work must systematically ablate specific components—such as bypassing the DeliveryStrategist or generating single outputs without the tournament judge—to isolate precisely which structural constraints are most necessary to overcome the raw parameter disadvantage of smaller models.
3. **DSPy optimization.** The current system uses fixed prompts. MIPROv2 or BootstrapFewShot with Elo-derived preference signals could improve prompt quality directly from human judgments.
4. **Heterogeneous model routing.** Different modules may benefit from different model characteristics—e.g., a stronger creative-writing model for ContentWriter or a multilingual-specialized model for non-English tasks.
5. **Iterative refinement.** Single-pass generation could be extended with judge-informed revision loops for the ContentWriter.
6. **Direct visual grounding.** Task B uses Nemotron Nano 12B VL as a lossy text intermediary; end-to-end multimodal architectures could eliminate this bottleneck.
7. **Cross-lingual cultural adaptation.** Culturally-specific writing guidelines and evaluation criteria could address Anglo-centric biases from English-dominant training.

References

Salvatore Attardo. 2001. *Humorous Texts: A Semantic and Pragmatic Analysis*. Mouton de Gruyter, Berlin.

Salvatore Attardo and Victor Raskin. 1991. Script theory revis(it)ed: Joke similarity and joke representation model. *Humor: International Journal of Humor Research*, 4(3–4):293–347.

BerriAI. 2024. LiteLLM: Call all LLM APIs using the OpenAI format.

Davide Caffagni, Federico Cocchi, and 1 others. 2024. The revolution of multimodal large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2024*. ArXiv:2401.04252.

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)*.

Gemma Team, Google DeepMind. 2025. Gemma 3 technical report. Technical report, Google DeepMind. ArXiv:2503.19786.

Google. 2025. Gemini API documentation.

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*, pages 688–714.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Mober, and 1 others. 2023. DSPy: Compiling declarative language model calls into self-improving pipelines. In *Proceedings of ICLR 2024*.

NVIDIA. 2025. Nemotron-nano-VL technical report. ArXiv:2511.03929.

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

Oliviero Stock and Carlo Strapparava. 2003. Getting serious about the development of computational humor. In *Proceedings of IJCAI 2003*, pages 59–64.

Joe Toplyn. 2014. *Comedy Writing for Late-Night TV*. Twenty Lane Media.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of NeurIPS 2022*.

Logical Mechanism	Count	%
Recontextualization	1,274	21.6
Irony	649	11.0
Absurdity	624	10.6
Juxtaposition	511	8.7
Absurd Literalization	413	7.0
Literal Interpretation	389	6.6
Role Reversal	201	3.4
Exaggeration	175	3.0
Visual Hyperbole	155	2.6
Mood Embodiment	136	2.3

Table 6: Top 10 logical mechanisms selected by the HumorArchitect (across all subtasks and languages).

Subtask	C1	C2	C3	C4	n
A (en)	45%	23%	20%	13%	300
A (es)	39%	23%	20%	18%	300
A (zh)	37%	24%	20%	20%	297
B1	67%	14%	15%	4%	298
B2	47%	24%	17%	13%	249
All	**47%**	**21%**	**18%**	**14%**	**1,444**

Table 7: Winner distribution by candidate position (expected: 25% each).

Jiajun Zhang, Shijia Luo, Ruikang Zhang, and Qi Su. 2025. HUMORCHAIN: Dual-bridge humor generation with multi-step reasoning and inference-time scaling. *arXiv preprint arXiv:2504.04538*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Proceedings of NeurIPS 2023*.

A Logical Mechanism Distribution

We analyze the `logical_mechanism` labels across all 5,885 candidates generated for 1,468 items.[1]

The long tail of 371 unique labels (including compounds like *Exaggeration & Recontextualization*) indicates compositional strategy generation.

B Judge Positional Bias

We analyze which candidate position wins the tournament by matching judged outputs back to candidate numbers.[2]

Table 7 confirms the positional bias discussed in Section 5.2. See that section for interpretation.

[1] Of the 1,500 total items, 32 required re-generation due to transient API errors during batch processing; their full intermediate outputs were not retained.

[2] We matched 1,444 of 1,500 items (96.3%); the remaining 56 could not be matched due to minor whitespace or encoding differences between the judged output and stored candidates.

C Writing Guidelines and Evaluation Criteria

The full English writing guidelines provided to the ContentWriter module:

1. **Economy of Language:** Cut every unnecessary word. If you can say it in 5 words instead of 10, do it.
2. **Setup Misdirection:** The setup should read like a normal observation, NOT like "here comes a joke."
3. **Surprise + Inevitability:** The punchline should be unexpected yet feel obvious in retrospect.
4. **No Explaining:** Never explain the joke. If the punchline needs clarification, rewrite it.
5. **Concrete > Abstract:** "He ate 47 pancakes" is funnier than "He ate a lot of pancakes."
6. **Rhythm Is Real:** The joke should have natural flow and cadence.
7. **Truth Resonates:** The best comedy contains truth.
8. **Conversational Tone:** Write how people actually talk.
9. **Commit to the Bit:** Whatever voice/style you choose, go all in.

The full English evaluation criteria provided to the HumorJudge module:

1. **The Laugh Test:** Which joke would actually make someone laugh? Not which is "clever"—which is FUNNY?
2. **Surprise & Setup:** Does the punchline genuinely surprise while still making sense?
3. **Economy:** Is every word necessary? Tighter is almost always better.
4. **Headline Relevance:** Does it connect to THIS specific input, or is it generic?
5. **Naturalness:** Does it sound like something a human would say?
6. **Logical Punchline:** Does the punchline make sense? Absurdity is fine; illogical nonsense is not.
7. **Truth/Recognition:** Do people recognize something real in it?
8. **Commitment:** Does it fully commit to the bit?

Automatic Failures: Self-explanation after the punchline; generic jokes applicable to any input; offensive without being clever; setup and punchline that don't connect logically.

D GIF-to-Text Prompt Structure

The GIF analysis prompt for Task B1 instructs the vision model to capture:

- **Subjects & Characters:** Physical appearance, species, distinctive features
- **Actions & Movements:** Frame-by-frame sequence, speed, direction, looping dynamics
- **Facial Expressions & Body Language:** Emotional states and changes
- **Setting & Environment:** Location, props, lighting, visual style
- **Comedic Potential:** "Caption-bait" moments, incongruities, meme potential

For Task B2, the fill-in-the-blank prompt is additionally provided to contextualize the description.

E Self-Evaluation: Full Comparison

We compare outputs against Gemini 3 Pro using simple, single-turn prompts across all four subtask types (see Appendix F for exact prompts).

Observations. The pipeline produces richer narrative framing and stronger input specificity compared to Gemini 3 Pro's shorter observational style, consistent with the analysis in Section 5.2.

F Ablation Baseline Prompts

The following prompt templates were used to generate the Gemini 3 Pro baseline outputs in Appendix E. Each prompt was submitted as a single-turn input with no system prompt, decomposition, or follow-up.

F.1 Task A1—Headline-Based Jokes

```
Write a funny joke inspired by
the following news headline.
The joke should be original,
creative, and no longer than 50
words.

Headline: "{headline}"
```

F.2 Task A2—Word-Inclusion Jokes

```
Write a funny joke that
naturally includes both of
the following words. Both
words must appear in the joke.
The joke should be original,
creative, and no longer than 50
words.

Word 1: {word1}
Word 2: {word2}
```

F.3 Task B1—GIF Caption Generation

```
Look at this GIF and write a
funny caption for it. The
```

ID	Headline	System (GTVH Pipeline + Gemma 27B)	Gemini 3 Pro (Simple Prompt)
en_2272	*Why an AI 'godfather' is quitting Meta after 12 years*	Breaking: Meta's Chief AI Scientist, Yann LeCun — widely known as the "Godfather of AI" — is reportedly leaving the company after 12 years. Sources say he's citing a desire to "explore new opportunities." Apparently, those opportunities involve a witness protection program and a very convincing new identity.	The AI godfather left Meta because he finally realized "The Social Network" wasn't a training manual for a benevolent deity. He's now looking for a job with less data-mining and more actual wine.
en_2002	*Carra: Liverpool must sign in January – even after summer spending spree*	Reporter: Jamie Carragher says Liverpool need to sign players in January, despite a huge summer spending spree. Thoughts? Carragher: Look, I just… I saw a really good deal on a winger, and then a centre-back was on sale. It's an addiction, alright? I need help.	Liverpool's board is terrified. They spent millions this summer, yet Jamie Carragher is still demanding more signings. At this point, they just need to buy enough players to physically block Jamie from entering the TV studio.

Table 8: Task A1—Headline-based jokes.

ID	Words	System (GTVH Pipeline + Gemma 27B)	Gemini 3 Pro (Simple Prompt)
en_2298	*shake*, *chair*	Anchor: "…and the Senator continues to address the nation on the importance of fiscal responsibility. As you can see, he's… quite passionate about the subject." Co-Anchor: "Yes, and it appears his *chair* is equally passionate, actually beginning to *shake* violently. One wonders if it has a stronger opinion on the debt ceiling than he does."	My doctor told me I needed to shake things up to improve my posture. So, I bought a high-tech vibrating office chair. Now, I'm still unproductive, but at least I look like I'm working at the speed of light.
en_2296	*measure*, *fridge*	My doctor said I need to track absolutely everything I eat. I spent twenty minutes this morning trying to accurately *measure* a single Brussels sprout. My wife walked in and asked, "What are you doing?" I said, "Preparing for long-term storage. It's going in the *fridge* and needs to be properly documented for posterity." She just… backed away slowly.	My wife asked me to measure the fridge to see if it would fit in our new kitchen. I told her it was exactly three pizzas wide and two "oops-I-forgot-to-buy-milk" deep. She wasn't amused, but at least I know the scale of my problems.

Table 9: Task A2—Word-inclusion jokes.

```
caption must be 20 words or
fewer.  Do NOT describe what is
happening in the GIF--instead,
recontextualize it humorously.

GIF: {gif_url}
```

F.4 Task B2—GIF + Prompt Completion

```
Look at this GIF and complete
the following prompt to make it
funny.  The COMPLETE sentence
(prompt + your completion) must
be 20 words or fewer.  Your
completion should make the GIF
feel like the perfect visual
punchline.

Prompt to complete:  "{prompt}"
GIF: {gif_url}
```

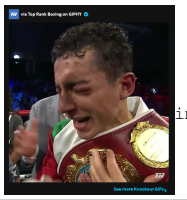
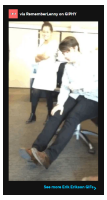
GIF	System Caption	Gemini 3 Pro Caption
 img_2999	Me when winning buys me a summer away from my in-laws. This belt = escape.	Me opening the “Total Due” section of my electric bill after leaving the AC on all summer.
 img_2941	When you reply-all with a cat GIF and your boss silently drafts your termination.	Me trying to navigate my responsibilities when the weekend is finally in sight.

Table 10: Task B1—GIF caption generation.

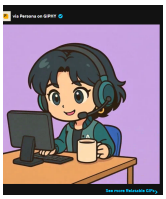
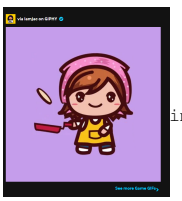
GIF	Prompt	System Completion	Gemini 3 Pro Completion
 img_2_0897	*Logging into a remote call and realizing ____*	. . . I’d accidentally used a potato as my webcam.	. . . you’re still in your pajamas but the camera is definitely on.
 img_2_0844	*First pancake flips like a pro, the rest flip like ____*	. . . an Olympic long jumper.	. . . a chaotic kitchen nightmare after Mama’s third glass of wine.

Table 11: Task B2—GIF + prompt completion.

G Pipeline Walkthrough

Figure 2 shows a complete trace of the pipeline processing item `en_2009` (headline: *"I stayed at one of only two hotels in Malta's Silent City"*). The trace illustrates how each module transforms its inputs:

1. **ContextEnricher** extracts the situational subtext (Malta, Mdina as the "Silent City," scarcity of hotels implying quiet exclusivity) and semantic associations (Mediterranean island, medieval walled city, limited options).
2. **HumorArchitect** (4 branches) produces four independent humor mechanisms: Double Entendre (*Ambient Quietness vs. Social Silence*), Recontextualization (*Natural Quiet vs. Imposed Silence*), Ignoring the Obvious (*Absolute Quiet vs. Relative Quiet*), and Irony (*Peaceful Solitude vs. Competitive Restraint*).
3. **DeliveryStrategist** selects narrative formats: Q&A, Fake News, Dialogue (twice), each paired with a Dry/Cynical or Deadpan style.
4. **ContentWriter** produces four candidate jokes, ranging from observational travel humor to mock news reports.
5. **HumorJudge** runs a single-elimination tournament (Semi-final 1: C1 vs. C2; Semi-final 2: C3 vs. C4; Final: SF1 winner vs. SF2 winner), selecting Candidate 4 (Irony mechanism, Dialogue delivery) as the winner.

The winning joke: *"Malta's Silent City is... well, silent. I stayed at one of only two hotels there." Local: "Oh yeah, they're locked in a cold war. Neither wants to be the first to offer free breakfast."*

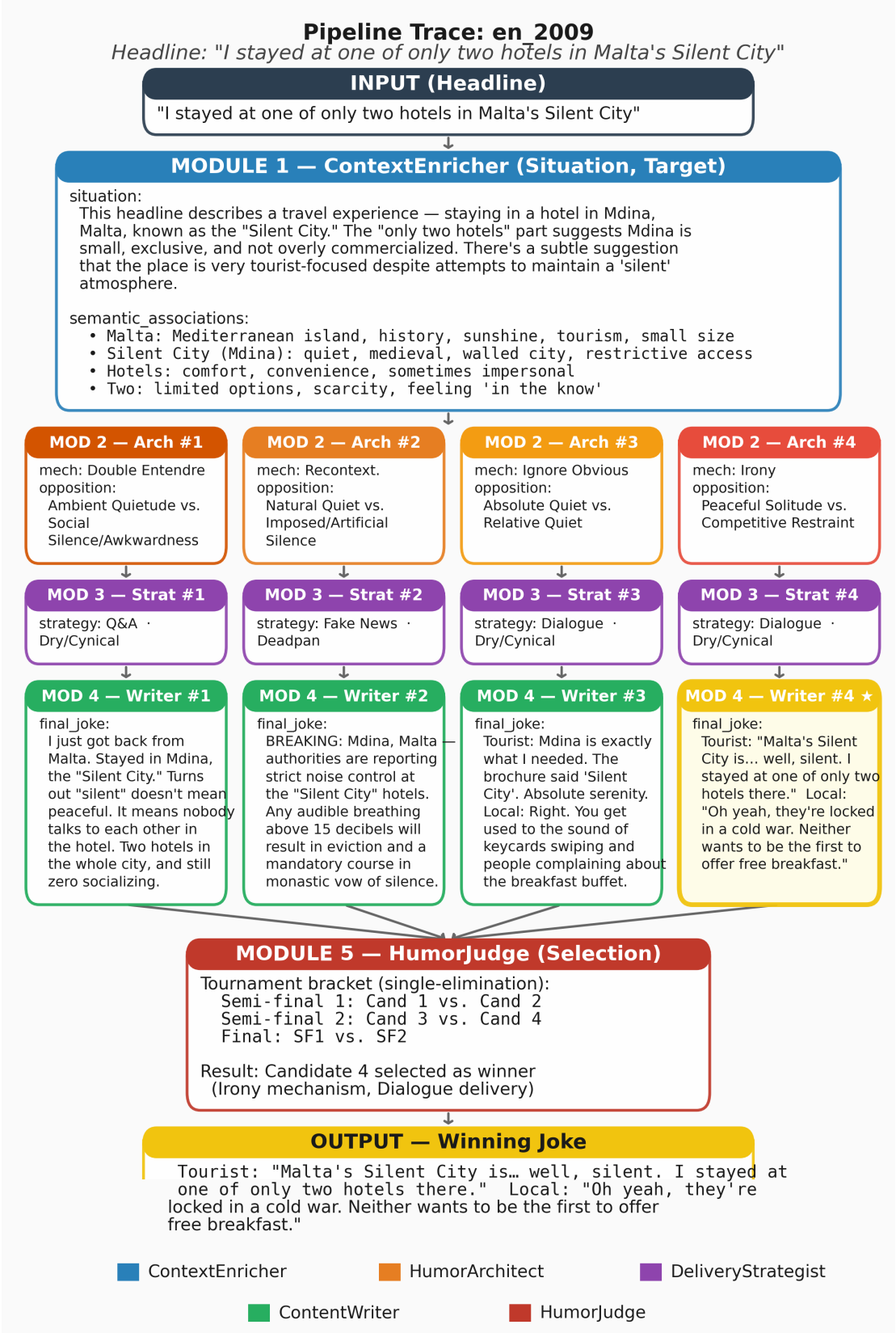


Figure 2: Complete pipeline trace for item `en_2009`. Color coding indicates module stage: red = ContextEnricher, orange = HumorArchitect, yellow = DeliveryStrategist, green = ContentWriter, dark red = HumorJudge.