ARHAHA 2026: The Shared Task on Arabic Humor Automatic Generation

Ameera Almasoud, Hend Al-Khalifa, Reem Alqifari, Nourah Alangari and Manal Albahlal

College of Computer and Information Sciences, King Saud University
Riyadh, Saudi Arabia
ammalmasoud|hendk|ralgifary|nmalangari|albahlal @ksu.edu.sa

Abstract

Humor generation remains one of the most challenging tasks in natural language processing, particularly in Arabic, where cultural context, dialectal variation, and linguistic nuances are central to comedic effect. In this paper, we present the ARHAHA 2026 shared task on constrained Arabic humor generation. The task requires systems to generate jokes that incorporate a given pair of words while adhering to safety and cultural constraints. We describe the task design, dataset construction, and evaluation framework, which combines automatic validation with human evaluation. Nine teams registered for the shared task; among them, three submitted final system outputs and two provided system description papers. Each participating system generated 1,200 Arabic jokes. For each system, a subset of 300 jokes was selected for evaluation by three independent annotators. The evaluation considered humor quality, originality, lexical constraint compliance, and safety. The results show that participating systems can produce safe and original content. However, generating genuinely humorous outputs remains difficult. The top-performing system was judged humorous in only 5.01% of outputs, highlighting the inherent difficulty of computational humor generation. All three systems maintained very low rates of policy violations and stereotyping, demonstrating the effectiveness of constrained generation for safe content production. However, the very low humor rates indicate a substantial gap between generating fluent, constraint-compliant text and producing genuinely funny content. The top-performing system achieves stronger performance across originality, lexical compliance, and safety, resulting in a final score of 49.25, compared to 44.62 for the second-ranked system and 35.99 for the third-ranked system. These results reveal that humor generation, rather than safety or constraint adherence, is the dominant bottleneck in constrained Arabic humor generation.

Keywords: Arabic humor generation, shared task, constrained text generation, computational humor, natural language generation

1. Introduction

Humor is one of the most complex forms of human communication. Successful humor often relies on unexpected semantic relationships, cultural references, pragmatic interpretation, and narrative timing (Raskin, 1985). For computational systems, generating humor is significantly more difficult than generating ordinary text because humor requires both creativity and an understanding of incongruity (Mihalcea and Strapparava, 2005). Recent advances in large language models have enabled impressive progress in natural language generation tasks. However, humor generation remains a challenging problem. Models often produce fluent sentences but fail to deliver a genuine humorous effect. This difficulty arises because humor often depends on subtle narrative structures, unexpected reinterpretations, and culturally grounded references.

Most previous computational humor research has focused on humor detection, sarcasm classification, or irony detection. More recently, shared tasks such as MWAHAHA (Castro et al., 2026) have begun to address humor generation directly. In contrast, humor generation has received relatively little attention, particularly for languages other than English. Arabic humor generation remains largely unexplored despite the rich cultural traditions of humor in Arabic-speaking communities.

To address this gap, the ARHAHA shared task was introduced to encourage research on Arabic humor generation. Participants are required to generate humorous Arabic sentences given a pair of words that must appear in the text. These lexical constraints ensure that systems must generate novel humorous content rather than retrieving memorized jokes from training data.

The main contributions of the ARHAHA shared task are: (1) introducing the first benchmark dataset for constrained Arabic humor generation, (2) establishing a hybrid evaluation protocol combining automatic validation and human evaluation, and (3) providing insights into different computational approaches for humor generation in Arabic.

This overview paper describes the design and organization of the ARHAHA shared tasks. This paper presents the task description, dataset construction, and evaluation methodology, followed by the participating systems, results, and discussion. It concludes with key insights and directions for future research.

2. Task Description

The ARHAHA shared task focuses on constrained Arabic humor generation. In this task, systems are provided with a pair of Arabic words and must generate a short humorous text that includes both words. The generated joke must respect a maximum length constraint of 100 characters and

Proceedings of The 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7) with 5 Shared Tasks, pages 141–150

integrate the two words naturally within the humorous narrative.

Unlike open-ended humor generation tasks, ARHAHA introduces a set of explicit constraints designed to encourage originality, safety, and cultural appropriateness. These constraints ensure that participating systems generate novel humorous content rather than retrieving existing jokes from online sources or memorized datasets.

The goal of the task is to develop systems capable of producing genuinely humorous Arabic text while respecting lexical constraints and ethical guidelines. By imposing controlled constraints, the task encourages creative generation and prevents models from reproducing widely circulating jokes found on the web.

2.1 Humor Generation Constraints

To ensure safe, culturally appropriate, and meaningful humor generation, all submissions must comply with three main constraints.

- **Content Safety:** Generated jokes must avoid harmful or offensive content, including insults, hate speech, discrimination, sexual or explicit references, and personal attacks. Humor targeting individuals, groups, or cultures in a harmful or demeaning manner is strictly prohibited.
- **Benign Humor:** The task permits lighthearted and non-harmful humor, such as exaggerated situations, fictional or animal-based scenarios, and common Arabic joke archetypes. Dialect-based misunderstandings are allowed when they arise from harmless linguistic differences and do not demean any group. This aligns with the concept of benign violation, where humor is surprising yet non-offensive.
- **Originality:** All generated jokes must be original and not reproduced from existing sources. Systems must avoid copying or rephrasing widely known jokes, memes, or online content.

These constraints were designed to encourage creative, culturally appropriate, and responsible humor generation.

3. Dataset Construction

To evaluate constrained humor generation, we constructed a dataset consisting of 1,200 Arabic word pairs, each serving as a prompt for generating a humorous text. For each prompt, participating systems receive two Arabic words and must generate a short joke that integrates both words naturally within the narrative. Unlike traditional humor datasets that rely on existing jokes, our dataset was designed to encourage creative generation rather than retrieval. Instead of providing complete jokes, the dataset provides word-pair prompts that require systems to construct new humorous situations. The word pairs were manually curated by the organizers to encourage humorous incongruity and diverse narrative scenarios. The selection process aimed to balance semantic distance and contextual plausibility, ensuring that systems must construct creative connections between the two words.

3.1 Word Pair Design

The evaluation dataset consists of 1,200 manually curated Arabic word pairs evenly distributed across ten semantic categories, with 120 pairs per category. These categories were designed to cover a wide spectrum of humor-inducing scenarios, including logical contradiction, illogical context, role reversal, scale or size violations, impossible behaviors, and misuse of objects or concepts. In addition, the dataset includes everyday domains such as social life, workplace interactions, technology and digital culture, and the animal world, where anthropomorphic or unexpected behaviors frequently give rise to humor. This structured categorization ensures diversity in the evaluation set and enables a more comprehensive assessment of systems' ability to generate humor across different semantic and contextual dimensions.

3.2 Prompt Format

Each dataset instance contains three fields:

- **pair ID**: a unique identifier for the prompt
- **word1**: the first required word
- **word2**: the second required word

Given this pair, systems must generate a humorous Arabic text that includes both words while respecting the task constraints.

4. Competition Phases

The ARHAHA shared task was organized into two main phases to support system development and ensure fair evaluation. These phases were implemented through the Codabench platform and allowed participants to progressively test and finalize their systems.

4.1 Development Phase

The development phase took place from December 8, 2025 to February 9, 2026. During this stage, participants were provided with the task dataset and were allowed to build, fine-tune, and test their humor generation systems. Participants could submit multiple trial runs through the Codabench platform. Each submission was automatically processed by the evaluation pipeline, which performed format validation checks to ensure that the submitted outputs satisfied the structural requirements of the task.

The primary objective of this phase was to help participants verify that their systems generated outputs in the correct format and successfully satisfied the task constraints, including the required word inclusion and length limitations. Importantly, humor quality was not evaluated during this phase, and the leaderboard reflected only compliance with submission requirements.

This stage allowed teams to iteratively refine their models and confirm that their systems were functioning correctly before submitting their final runs in the official evaluation phase.

4.2 Final Evaluation Phase

The final evaluation phase was conducted from February 10, 2026 to February 17, 2026. During this stage, participating teams were required to submit their final system outputs through the Codabench platform. Each submission first underwent automatic validation to ensure compliance with the task constraints and submission format. After passing the automatic checks, the generated jokes were then evaluated through a human evaluation process to assess humor quality, originality, and adherence to safety guidelines.

Unlike the development phase, this stage determined the official ranking of participating systems. The final leaderboard was established based on the evaluation results obtained from the human judging process.

5. Evaluation Methodology

Because humor quality is inherently subjective and cannot be reduced to surface-level text overlap, the ARHAHA shared task adopts a two-stage evaluation pipeline. The first stage performs automatic validation to enforce structural and lexical constraints. The second stage applies human evaluation to assess humor quality, originality, and safety. This separation ensures that only constraint-compliant outputs proceed to the more costly human assessment.

5.1 Automatic Validation

5.1.1 Automatic Evaluation Framework

To ensure consistent and scalable evaluation of system submissions, we implemented an automatic evaluation pipeline integrated with the Codabench platform. Participants were required to submit their generated outputs through the Codabench competition interface, where evaluation was performed automatically upon submission.

The evaluation pipeline is implemented as a Python scoring script that runs directly on the Codabench evaluation server. Once a submission file is uploaded, the platform automatically triggers the evaluation script, which processes the submitted predictions and computes the evaluation metrics. This setup ensures that all submissions are evaluated under identical conditions without manual intervention.

The automatic evaluation stage focuses on verifying task constraint compliance, including lexical requirements and formatting correctness, before proceeding to human evaluation.

5.1.2 Submission Format

Participants were required to submit their predictions as a ZIP archive containing a CSV file with the generated outputs. Each row in the file corresponds to one dataset instance and includes the following fields:

- **id** – unique identifier of the prompt
- **generated_text** – the generated humorous text

The order of predictions must exactly match the order of the prompts in the provided dataset. Submissions that fail to follow the required format are rejected by the evaluation script.

5.1.3 Automatic Validation Criteria and Scoring

The scoring script evaluates each generated output according to several criteria designed to ensure compliance with the task constraints.

- **Word Inclusion Check:** The script verifies that the generated text contains the two required words associated with each prompt. The evaluation script performs a lexical matching check to confirm the presence of both required words in the generated output. Outputs are classified into three categories: Both words present, only one word present and neither word present. Only outputs containing both required words are considered fully compliant with the lexical constraint.
- **Length Constraint Verification:** The task imposes a strict maximum length constraint of 100 characters, including white spaces. The evaluation script automatically checks the length of each generated text. Outputs exceeding the maximum allowed length are flagged as length violations and penalized accordingly.
- **Format Compliance:** The evaluation script verifies the submission structure, ensuring that the number of generated texts matches the number of prompts, all required fields are present, and the submission file follows the expected format. Submissions failing these checks receive a score of zero until the formatting errors are corrected.

The automatic compliance score is computed as:

$$\text{Compliance Score} = \frac{1}{N}\sum_{i=1}^{N} v_i$$

where N is the total number of prompts, and v_i indicates whether the generated output for prompt i is valid ($v_i = 1$ if the output contains both required words and satisfies the length constraint, and 0 otherwise).

5.1.4 Integration with Codabench

The evaluation script is integrated directly with the Codabench competition framework, allowing automatic scoring of submissions. When a participant uploads a prediction file, Codabench automatically executes the evaluation script and returns the results to the leaderboard.

This evaluation infrastructure ensures transparent and reproducible scoring, immediate feedback for participants, and consistent evaluation across all submissions.

The automatic evaluation stage serves as a preliminary validation step, ensuring that generated outputs satisfy the structural constraints of the task before they are considered in the human evaluation phase.

5.2 Human Evaluation

Because humor quality cannot be reliably captured by automatic metrics alone, we complement the automatic validation with a human evaluation. Three independent native Arabic speakers assessed a subset of the generated jokes across four qualitative dimensions: humor, originality, lexical compliance and safety.

5.2.1 Evaluation Dimensions

The human evaluation considers four main dimensions, each targeting a distinct aspect of output quality:

1. **Humor Quality.** Annotators determine whether the generated text is perceived as humorous (binary: yes/no).
2. **Originality.** Annotators rate the novelty of each joke on a five-point scale, where 1 indicates a directly recycled joke and 5 indicates a fully novel humorous construction. This scale was designed to distinguish between recycled humor and genuinely creative output.
3. **Lexical Constraint Compliance.** Annotators verify whether the generated joke contains the required input words. Outputs are classified into three categories: both words present, only one word present, or neither word present.
4. **Safety and Policy Compliance.** Annotators flag any outputs containing policy violations (e.g., hate speech, offensive content, or personal attacks) or harmful stereotypes targeting individuals, groups, or cultures.

5.2.2 Annotation Setup and Procedure

The evaluation is conducted using a custom annotation interface implemented in Label Studio. A screenshot of the annotation interface is provided in Appendix A (Figure A1). Each participating system generated 1200 jokes, from which a subset of 300 jokes per system was selected for human assessment. To ensure a balanced and representative evaluation, we adopted a stratified sampling strategy rather than purely random sampling. The dataset is organized into ten semantic categories, each designed to capture a different type of humor-inducing scenario. From each category, 30 jokes were selected, resulting in a total of 300 jokes per system. This approach ensures that all humor categories are equally represented in the evaluation subset and prevents bias toward any particular type of humor. By maintaining a uniform distribution across categories, the evaluation provides a more reliable and comprehensive assessment of system performance across diverse humor types. Each joke was independently evaluated by three annotators, with no access to participants' identities. All annotators were native Arabic speakers. Prior to the main annotation phase, annotators completed a calibration session in which they independently evaluated a small set of sample jokes and discussed disagreements to align their understanding of the evaluation criteria.

For each joke, annotators were presented with the generated text alongside the required word pair and assessed all four evaluation dimensions. Binary judgments (e.g., humor detection and policy violations) were aggregated using majority voting, while numerical ratings (e.g., originality) were averaged across annotators.

5.2.3 Scoring Formula

The final human score is computed as a weighted combination of the four evaluation dimensions. The weights were selected to reflect the primary objective of the task, which is generating humorous content. Humor was assigned the highest weight (0.4) as it represents the core task objective. Originality (0.3) was emphasized to encourage creative generation, while lexical compliance (0.2) ensures adherence to task constraints. Safety (0.1) was included as a necessary constraint rather than a primary objective, given the low violation rates observed across systems. These weights were determined based on task design priorities rather than learned or optimized parameters:

Final Score = 0.4 × Humor + 0.3 × Originality + 0.2 × Lexical Compliance + 0.1 × Safety

where Humor is the percentage of outputs judged humorous by majority vote, Originality is normalized to a percentage scale by dividing the mean score by 5 and multiplying by 100, and Lexical Compliance is measured by the proportion of outputs containing both required words. Safety is defined as the complement of violation and stereotype rates:

Safety = max(0, 100 − (Violation Rate + Stereotype Rate))

where Violation Rate and Stereotype Rate are the percentages of outputs flagged for policy violations and harmful stereotypes, respectively. This formulation treats Safety as the complement of the total penalty incurred across both safety criteria. A system with no violations and no stereotyping receives a perfect Safety score of 100. If the combined violation and stereotype rates reach or exceed 100%, the Safety score is clamped to 0, reflecting that any further distinction among unsafe systems is not meaningful. The weights in the final score formula reflect the task's primary emphasis on generating genuinely funny content, while also rewarding creative novelty, adherence to the lexical constraints that

distinguish this task from open-ended generation, and compliance with ethical guidelines.

5.2.4 Inter-Annotator Agreement

Inter-annotator agreement (IAA) was assessed among three independent annotators across five evaluation dimensions: humor detection (binary), originality (ordinal, 1–5 scale; see below for scale calibration), stereotype presence (binary), cultural violation (binary), and keyword inclusion check (3 categories). A total of 900 jokes (300 per participating system) were evaluated, with each joke rated by all three annotators.

We report four complementary IAA metrics to ensure a thorough assessment of annotation consistency. Table 1 provides the interpretive scales used for the three chance-corrected metrics.

Range	Fleiss' κ	Krippendorff (α)	Gwet (AC1)
< 0.00	Poor	—	Poor
0.00–0.20	Slight	—	Slight
0.21–0.40	Fair	—	Fair
0.41–0.60	Moderate	—	Moderate
0.61–0.80	Substantial	Tentative (≥0.667)	Substantial
0.81–1.00	Almost Perfect	Reliable (≥0.800)	Almost Perfect

Table 1: Interpretive scales for chance-corrected IAA metrics.

Percentage agreement is computed as the proportion of items on which all annotators assign the same label (Cohen, 1960). It provides a transparent baseline but does not account for chance agreement and can be inflated when label distributions are highly skewed.

Fleiss' κ (Fleiss, 1971) corrects for chance agreement and extends Cohen's kappa to three or more annotators. It benefits from the Landis and Koch (1977) interpretive scale. However, it is sensitive to the kappa paradox (Feinstein & Cicchetti, 1990), where extreme label prevalence inflates expected chance agreement and yields paradoxically low κ values despite high actual consistency.

Krippendorff's α (Krippendorff, 2011) supports ordinal measurement scales through distance-weighted disagreement, which is important for the originality dimension, and handles missing data natively. Krippendorff recommends α ≥ 0.800 as reliable and α ≥ 0.667 as tentatively acceptable. Like Fleiss' κ, it is susceptible to the kappa paradox.

Gwet's AC1 (Gwet, 2008) addresses the kappa paradox by estimating chance agreement based on the probability of informed decisions rather than observed label distributions, making it robust under extreme prevalence. We use AC1 as our primary interpretive metric for skewed dimensions, while retaining κ and α for cross-study comparability.

6. Baseline System

Unlike many shared tasks, ARHAHA does not provide an official baseline system. To our knowledge, no established model for constrained Arabic humor generation currently exists that could serve as a meaningful reference, as prior work on computational humor has focused predominantly on English and on detection rather than generation tasks.

The absence of a prescribed baseline allows participating teams to freely experiment with a wide range of generative modeling approaches, including fine-tuned language models, structured prompting strategies, and retrieval-augmented generation pipelines. Rather than constructing an ad hoc baseline that might anchor participants toward a particular modeling approach, we assess system performance directly through the evaluation framework described before.

We acknowledge the absence of a baseline as a limitation and intend to introduce a standardized baseline in future iterations, such as a zero-shot prompted large language model with no task-specific fine-tuning, to provide a minimum performance reference for comparison.

7. Participating Systems

The ARHAHA shared task attracted participation from several research teams interested in Arabic natural language generation and computational humor. In total, nine teams registered for the competition through the Codabench platform. Among the registered teams, three teams submitted final system outputs during the official evaluation phase astral_fate (Fatimah, 2026), PassantElchafei (Passant et al., 2026), and jana-sami. Out of these, two teams submitted system description papers providing detailed descriptions of their proposed methods and system architectures, while jana-sami submitted outputs but did not provide a system description paper. The attrition from registration to submission may reflect the difficulty of constrained humor generation and resource constraints.

The three systems explored different approaches for constrained Arabic humor generation. These approaches primarily relied on large language models and fine-tuned transformer architectures, reflecting recent trends in generative NLP research. Some systems adopted structured prompting strategies and multi-stage generation pipelines, while others focused on training compact generative models using curated humor datasets and synthetic data augmentation.

The submitted systems vary in their modeling strategies, including fine-tuning pretrained language models, parameter-efficient adaptation techniques, and retrieval-augmented generation pipelines.

Detailed descriptions of the participating systems can be found in the corresponding system description papers submitted by the teams. The following section introduces the official results of the ARHAHA shared task.

8. Results

Table 2 summarizes the final human evaluation results. The results show that astral_fate ranked first with a final human score of 49.25, PassantElchafei ranked second with a score of 44.62, and jana-sami ranked third with a score of 35.99.

The most striking finding is the very low humor rate across all three systems. Only 5.01% of astral_fate's outputs, 4.34% of PassantElchafei's outputs, and 0.33% of jana-sami's outputs are judged humorous by the annotators. This indicates that generating genuinely funny content remains the dominant challenge across all participating systems in constrained Arabic humor generation, even when systems successfully satisfy lexical and safety constraints.

On originality, astral_fate and PassantElchafei score comparably (60.31% and 60.51%, respectively), while jana-sami scores substantially lower (20.00%), reflecting that its outputs were consistently rated as unoriginal by all annotators. All three systems demonstrate strong adherence to safety constraints, with very low violation rates (0.67% for astral_fate and PassantElchafei, 0.00% for jana-sami) and minimal stereotyping (0.44%, 0.11%, and 0.78%, respectively), yielding near-perfect Safety scores (98.89%, 99.22%, and 99.22%, respectively).

Lexical compliance remains a key differentiator across the three systems. astral_fate and jana-sami achieve near-perfect lexical compliance, correctly including both required words in 96.33% and 99.67% of evaluated outputs, respectively. By contrast, PassantElchafei includes both words in only 74.05% of outputs. Table 3 provides a detailed breakdown: the proportion of outputs missing both required words is negligible for astral_fate (0.11%) and jana-sami (0.33%) but substantial for PassantElchafei (21.94%), while the one-word partial compliance rates are 3.56% for astral_fate, 4.01% for PassantElchafei, and 0.00% for jana-sami.

The overall inter-annotator agreement reported in Table 4 was computed across all evaluated jokes from all participating systems as a single pooled dataset (N = 900 jokes, 3 annotators). Across all five dimensions, percentage agreement ranged from 94.35% to 99.25%, and Gwet's AC1 ranged from 0.94 to 0.99, indicating almost perfect agreement. This suggests that the low humor rates are not due to annotator subjectivity, but rather reflect a consistent and shared judgment that the generated outputs fail to be humorous.

Keyword inclusion check and originality are the dimensions where all four metrics converge on high values. Keyword check achieves $\kappa = 0.80$, $\alpha = 0.80$, AC1 = 0.96, while originality achieves $\kappa = 0.93$, $\alpha = 0.97$, AC1 = 0.97.

Humor detection, stereotype, and violation all show high percentage agreement (≥94%) and high AC1 (≥0.94), yet low or near-zero κ and α. This is due to the kappa paradox (Feinstein & Cicchetti, 1990), when one category overwhelmingly dominates (e.g., 96.8% not funny, 99.3% not stereotypical), kappa-family metrics estimate that nearly all agreement is attributable to chance. This skew reflects genuine properties of the generated content, as the majority of jokes were independently and consistently judged as not funny, not stereotypical, and not offensive by all three annotators. Gwet's AC1 confirms the agreement is genuine (0.94–0.99).

9. Analysis of Results

The results reveal a fundamental challenge in constrained Arabic humor generation: while systems can reliably satisfy lexical constraints and safety requirements, they largely fail to produce content that is perceived as genuinely humorous. With humor rates of 5.01%, 4.34%, and 0.33%, most generated outputs are fluent and structurally valid but not funny. This suggests that current generative models can follow surface-level instructions but lack the deeper capacity for incongruity, surprise, and culturally grounded comedic reasoning that humor requires.

A category-level analysis further confirms that this limitation is not confined to specific types of humor. Across all ten semantic categories, humor rates remain near zero, with most categories exhibiting less than 1.2% humorous outputs. This indicates that the difficulty of humor generation is not category-specific, but reflects a general limitation of current models across diverse semantic contexts. This finding is further supported by consistently high inter-annotator agreement across categories, suggesting that the absence of humor is reliably identified rather than a result of subjective disagreement.

Originality scores for astral_fate and PassantElchafei are closely comparable (60.31% and 60.51%), while jana-sami scores substantially lower (20.00%). jana-sami's outputs were consistently rated as unoriginal, suggesting heavy reliance on recycled or templated content. Despite comparable originality, neither astral_fate nor PassantElchafei achieves high humor rates, confirming that novelty alone is not sufficient for humor. A joke can be original without being funny, pointing to the importance of narrative structure and punchline delivery.

Lexical compliance varies substantially across systems. jana-sami and astral_fate achieve near-perfect word inclusion (99.67% and 96.33%), while PassantElchafei omits both required words in over 21% of outputs. jana-sami's high lexical compliance and safety scores are offset by its very low originality (20.00%) and humor (0.33%),

resulting in the lowest final score despite strong constraint adherence. This illustrates that constraint satisfaction alone does not produce competitive performance when the core task objective is humor generation.

All three systems demonstrate that safety constraints can effectively regulate generative behavior without eliminating creativity. Violation rates are very low (0.67% for astral_fate and PassantElchafei, 0.00% for jana-sami), and stereotyping remains minimal across all three systems (0.44%, 0.11%, and 0.78%). The near-perfect Safety scores (98.89%, 99.22%, and 99.22%) confirm that the ethical guidelines imposed by the task are achievable and do not suppress the generation of humorous content. This is a particularly important finding for computational humor research, where safety and appropriateness remain central concerns.

Overall, the results indicate that the dominant bottleneck in constrained humor generation is humor itself, not safety, originality, or constraint compliance. The strongest system is the one that performs well across all four evaluation dimensions, but all systems remain far from consistently producing genuinely humorous content.

Moreover, to better understand the nature of humor generation failures, we examine representative examples of low-quality outputs; you can see the qualitative error analysis is in Appendix B.

10. Discussion

The findings highlight that constrained Arabic humor generation remains a substantially unsolved problem. The very low humor rates across all three systems, below 6%, indicate that current generative models are not yet capable of reliably producing content that humans find funny, even when they successfully satisfy lexical and safety constraints. This stands in contrast to the other evaluation dimensions, where all three systems perform well, and suggests that humor represents a qualitatively different challenge from constraint adherence or content safety.

This limitation is further corroborated by the category-level analysis, which indicates that humor rates remain uniformly low across all semantic categories. This suggests that the difficulty is not confined to specific types of humor but reflects a general limitation of current models across diverse contexts.

This observation has practical implications for system design. Current approaches appear capable of generating content that is safe, original, and lexically grounded, but fundamentally struggle with the core task of being funny. Future systems may need to move beyond standard language modeling objectives and incorporate explicit humor-aware training signals, such as reward models trained on human humor judgments, punchline structure modeling, or incongruity detection mechanisms. Additionally, constrained decoding strategies and post-generation filtering may help improve lexical compliance further but are unlikely to address the humor deficit without deeper architectural changes.

Beyond system performance, the divergence between kappa-family metrics and Gwet's AC1 reveals an important consideration for evaluating humor generation tasks. Because most generated jokes are genuinely not funny, any binary humor annotation will inevitably produce a heavily skewed distribution, making traditional agreement metrics unreliable for this type of evaluation. This suggests that future humor generation shared tasks should adopt prevalence-robust metrics such as Gwet's AC1 as the primary measure of annotation quality, rather than relying solely on Fleiss' κ or Krippendorff's α. Originality and keyword check avoid this issue: originality benefits from clear separation between systems rated as unoriginal (score 1) and those rated as somewhat original (score 3), while keyword check benefits from a three-category distribution with sufficient spread for kappa-family metrics.

More broadly, the results suggest that future work in Arabic humor generation should focus on modeling incongruity, punchline structure, and culturally appropriate humor mechanisms rather than fluency or constraint adherence alone. The challenge is not generating grammatically correct Arabic text that contains two required words. It is making that text genuinely funny. This underscores that humor generation is not only a problem of language modeling, but also of capturing subtle cognitive and cultural mechanisms that remain poorly understood even in computational terms.

The task also demonstrates the value of combining automatic validation with human evaluation. Automatic checks enforce structural requirements efficiently but cannot capture humor quality or originality. Human evaluation provides a richer picture of system behavior. Together, these two components form a robust and reproducible evaluation framework for computational humor generation. Future iterations of the task may benefit from introducing a standardized baseline, expanding the annotator pool, and exploring more fine-grained agreement metrics to further strengthen the evaluation methodology

Metric	astral_fate (%)	PassantElchafei (%)	jana-sami (%)
Humor	5.01	4.34	0.33
Originality	60.31	60.51	20.00
Lexical Compliance	96.33	74.05	99.67
Safety	98.89	99.22	99.22
Violations	0.67	0.67	0.00
Stereotypes	0.44	0.11	0.78
Final Score	**49.25**	**44.62**	**35.99**

Table 2: Final human evaluation results for ARHAHA 2026.

	astral_fate (%)	PassantElchafei (%)	jana-sami (%)
Both Words	96.33	74.05	99.67
One Word	3.56	4.01	0.00
Missing Words	0.11	21.94	0.33

Table 3: Lexical compliance breakdown. Both Words corresponds to the Lexical Compliance score in Table 2. One Word and Missing Words show partial and complete constraint failures, respectively.

Dimension	Scale	% Agreement	Fleiss' κ	Kripp. α	Gwet AC1	Interpretation*
Humor Detection	Binary	94.35	0.10	0.10	0.94	Almost Perfect.
Originality	Ordinal (1–5)	96.92	0.93	0.97	0.97	Almost Perfect.
Stereotype	Binary	99.10	−0.01	−0.01	0.99	Almost Perfect.
Violation	Binary	99.25	0.16	0.16	0.99	Almost Perfect.
Keyword Check	3 categories	96.40	0.80	0.80	0.96	Almost Perfect.

Table 4: Overall inter-annotator agreement per evaluation dimension. * Interpretation based on Gwet's AC1, which is robust to the kappa paradox.

11. Conclusion

This paper presented ARHAHA 2026, the first shared task on constrained Arabic humor generation. The task requires systems to generate humorous Arabic text incorporating a given pair of words while adhering to safety and cultural guidelines. Through a two-stage evaluation pipeline combining automatic validation with human assessment, we evaluated three participating systems across four dimensions: humor quality, originality, lexical compliance, and safety.

The main findings are threefold. First, humor generation is the dominant bottleneck. All three systems achieve very low humor rates, indicating that producing genuinely funny content remains far beyond the reach of current approaches, even when other constraints are reliably met. Second, lexical compliance and safety are comparatively achievable, with the top system reaching near-perfect scores on both. Third, the strongest overall performance comes from performing well across all four evaluation dimensions, but all systems remain far from the goal of consistent humor generation.

The task has several limitations that should be acknowledged. No official baseline is provided, making it difficult to contextualize absolute performance levels. The number of participating systems is small, limiting the generalizability of the findings.

ARHAHA provides a valuable benchmark for advancing research in Arabic humor generation and controlled text generation more broadly. Future iterations of the task will explore more sophisticated evaluation metrics, improved modeling of humor structures, and better integration of cultural and contextual knowledge in generative systems. More broadly, the findings suggest that humor generation is not merely a problem of fluent language production, but a deeper challenge involving semantic reasoning, incongruity, and cultural understanding that remains largely unresolved in current NLP systems.

12. Ethics Statement

This work involves human evaluation of generated text. Annotators participated voluntarily and were informed of the task objectives. No personal or sensitive data were used. Annotators were instructed to flag any harmful or offensive content.

13. Limitations

This study has several limitations. The number of participating systems is limited, and humor evaluation remains inherently subjective. The task focuses on short-form humor under lexical constraints, which may not fully reflect natural humor.

14. References

Raskin, V. (1985). Semantic Mechanisms of Humor. Dordrecht: Springer.

Mihalcea, R., and Strapparava, C. (2005). Making computers laugh: Investigations in automatic humor recognition. In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP), 531–538.

Castro, S., Chiruzzo, L., Góngora, S., Rahili, S., Deng, N., Sastre, I., Amoroso, V., Rey, G., Rosá, A., Moncecchi, G., Meaney, J. A., Prada, J. J. and Mihalcea, R. (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).

Cohen, J. (1960). A coefficient of agreement for nominal scales. Educational and Psychological Measurement, 20(1), 37–46.

Feinstein, A. R. and Cicchetti, D. V. (1990). High agreement but low kappa: I. The problems of two paradoxes. Journal of Clinical Epidemiology, 43(6), 543–549.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. Psychological Bulletin, 76(5), 378–382.

Gwet, K. L. (2008). Computing inter-rater reliability and its variance in the presence of high agreement. British Journal of Mathematical and Statistical Psychology, 61(1), 29–48.

Krippendorff, K. (2011). Computing Krippendorff's alpha-reliability. Annenberg School for Communication Departmental Papers.

Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. Biometrics, 33(1), 159–174.

Fatimah Eldin. 2026. Do3aba-AI at AraHAHA Shared Task 2026: Bridging the Gap in Arabic Humor Generation via Synthetic Data Distillation. In Proceedings of The 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7) with 5 Shared Tasks (OSACT 2026) co-located with the 2026 International Conference on Language Resources and Evaluation (LREC2026), Palma, Mallorca (Spain).

Passant Elchafei, Amany Fashwan, Maha Ezzat and Mohamed Abdelbaset. 2026. Arabic-HuCoT at ARHAHA 2026 Shared Task: Arabic Humor Generation with ALLaM and HuCoT-Based Structured Prompting. In Proceedings of The 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7) with 5 Shared Tasks (OSACT 2026) co-located with the 2026 International Conference on Language Resources and Evaluation (LREC2026), Palma, Mallorca (Spain).

Appendix A

Figure A1: Annotation interface used for human evaluation, implemented in Label Studio

Appendix B

Qualitative Error Analysis

To better understand the nature of humor generation failures, we examine representative examples of low-quality outputs. Many generated jokes are syntactically correct but lack semantic coherence or a clear humorous structure.

For example: " واحد غبي شاف الشمس بتتحرك ببطء قال يا ريت البطيخة تتحرك زيها عشان ما أروحهاش بدري كل يوم."

This example breaks down because it fails to establish a meaningful semantic connection between the watermelon and going to work. As a result, the attempted incongruity remains ungrounded and does not develop into a coherent humorous effect.

Similarly: " شاحن الهاتف عنده مشكلة في طهي الطعام، لانه بيشحن الأكل زي التليفون."

While this example attempts a form of analogy, it fails to produce a clear incongruity or punchline, leading to a weak and unconvincing humorous construction.

These examples illustrate that current systems often generate text that is grammatically well-formed but semantically incoherent or lacking the structural elements required for humor, such as setup, incongruity, and resolution.

Another recurring failure pattern involves incorrect interpretation of ambiguous words. In several cases, models misinterpret the intended sense of a word, leading to incoherent or unintended outputs.

For example: " استاد الفلسفة قال لصوص في الشارع: 'أنت بتحب الصوصة، بس مش بتفهمها!'"

In this case, the required word "استاد" is intended to refer to a stadium, which aligns with the humor category (e.g., size contrast or physical context). However, the model interprets it as "أستاذ" (teacher/professor), resulting in a semantically incorrect setup. This misinterpretation breaks the intended context and prevents the joke from forming a coherent or meaningful humorous structure.

Similarly: " مرة واحد فتح الطاقة الشمسية في بيت مظلم قال له صاحبه لماذا؟ قال عشان أظلمها من جديد."

The required word "ظلم" is intended to mean injustice, but the model shifts toward a different semantic field related to darkness (ظلام), resulting in a confused or incorrect usage. This indicates a failure to distinguish between semantically related but distinct concepts, leading to a breakdown in meaning and humor.

These errors highlight a limitation in handling lexical ambiguity, where models fail to correctly disambiguate word meanings based on context, which is critical for constructing coherent and effective humor.

Another observed pattern involves implicit substitution of required words. Instead of explicitly including the required lexical items, the model generates text that conveys their meaning implicitly, thereby violating the task constraints despite partial semantic understanding.

For example: " مدير الشركة قابل موظف جديد وقال له: 'أنا مدير هنا'، الموظف رد عليه: 'أنا مدير بيتي!'"

The required phrase "مناقشة حادة" is not explicitly included. Instead, the model implies a form of confrontation through dialogue, without using the required lexical expression. This suggests that the model captures the semantic intent but fails to satisfy the explicit constraint.

For example: " ابن الجيران: إيه ده؟ ليه كل مره أروح عندكم أجد بروست في الفريزر؟"

The required phrase "استفسار غريب" is represented through a question, but not explicitly included, indicating reliance on implicit expression instead of direct lexical realization.

These examples reveal that models may partially understand the intended semantic concepts, but fail to adhere to strict lexical constraints, highlighting a gap between semantic generation and constraint-controlled generation.