UCO-UC-CICESE-Plenitas Team at HAHA: Humor Analysis based on Human Annotation and Automatic Humor Generation using Embeddings and Large Language Models

Yoan Martínez-López[1,2,*], Ireimis de las Mercedes Leguen de Varona[3], Miguel Bethencourt[3], Julio Madera[3], Ansel Rodríguez-González[4], Kristell Aguilar Alarcón[1,2], José Carlos Arévalo Fernández[2], Carlos de Castro Lozano[1,2] and Jose Miguel Ramirez Uceda[1,2]

[1]*EATCO. Universidad de Córdoba, Córdoba, Spain*
[2]*Plénitas, C/ Le Corbusier s/n, 14005 Córdoba, Spain*
[3]*Universidad de Camaguey, Circunvalación Norte, Camino Viejo Km 5 y 1/2, Camaguey, Cuba*
[4]*Centro de Investigación Científica y de Educación Superior de Ensenada, Unidad Académica Tepic, México*

Abstract

This paper describes the participation of the UC-UCO-CICESE-Plenitas team in the HAHA 2026 shared task (Humor Analysis based on Human Annotation and Automatic Humor Generation), organised within IberLEF 2026. This edition moves beyond the traditional detection-only setting and proposes three subtasks: humor detection (classifying a news headline as satirical or real), LLM-generated humor detection (distinguishing jokes written by humans from those produced by a large language model), and humor generation from headlines, evaluated through human-preference judgments in arena-style pairwise battles and ranked on an Elo leaderboard. We propose a single configurable pipeline that routes each subtask to the appropriate strategy. For the two classification subtasks we use a two-component ensemble: a multilingual sentence-embedding classifier (`intfloat/multilingual-e5-large-instruct` as the primary encoder, with a logistic-regression or k-nearest-neighbour head) and a large language model served through Ollama operating zero-shot, whose probability distributions are merged by a weighted average. For generation we adopt a retrieval-augmented generation (RAG) pipeline in which human-authored jokes retrieved by embedding similarity are inserted as few-shot examples to condition generation. In the evaluation phase our system ranks second of eight on Subtask 1 ($\mathrm{F1} = 0.9525$), outperforming the official baseline by more than 0.33 F1, and obtains $\mathrm{F1} = 0.7333$ on Subtask 2. Our system ranked first in Task 3 with a rating of 1136, statistically tied with the next-best system and significantly outperforming the official baseline by 301 rating points. The contrast between the three subtasks is our most informative finding: the same system that reliably discriminates satirical from real headlines struggles to separate human-written from machine-generated humor, an intrinsically subtler distinction.

Keywords

computational humor, humor detection, humor generation, multilingual embeddings, large language models, ensemble learning, retrieval-augmented generation, Spanish NLP, HAHA

1. Introduction

Humor is one of the most distinctively human forms of communication, and although it has long been studied from psychological, cognitive, and linguistic perspectives, its computational treatment remains a notoriously hard and only partially solved problem. Within Machine Learning and Computational Linguistics, the automatic study of humor has gained sustained traction over the last two decades, beginning with foundational work on humor recognition [1, 2, 3, 4, 5, 6, 7, 8] and maturing through a

IberLEF 2026, September 2026, León, Spain
*Corresponding author.
yoan.martinez@plenitas.com (Y. Martínez-López); ireimis.leguen@reduc.edu.cu (I. d. l. M. L. d. Varona); betamiguel00@gmail.com (M. Bethencourt); julio.madera@reduc.edu.cu (J. Madera); ansel@cicese.edu.mx (A. Rodríguez-González); kristell.aguilar@plenitas.com (K. A. Alarcón); josecarlos@plenitas.com (J. C. A. Fernández); carlosdecastrolozano@gmail.com (C. d. C. Lozano); p52raucj@uco.es (J. M. R. Uceda)
0000-0002-1950-567X (Y. Martínez-López); 0000-0002-1886-7644 (I. d. l. M. L. d. Varona); 0000-0001-8873-6802 (M. Bethencourt); 0000-0001-5551-690X (J. Madera); 0000-0001-9971-0237 (A. Rodríguez-González); 0000-0001-8913-8388 (K. A. Alarcón); 0009-0004-4155-957X (J. C. A. Fernández); 0000-0001-6603-843X (C. d. C. Lozano); 0000-0002-5027-7521 (J. M. R. Uceda)

series of shared tasks. Yet progress has concentrated overwhelmingly on the *detection* and *classification* of humor, while a characterisation rich enough to support its automatic *recognition* and *generation* is still far from being achieved. The difficulty is compounded by the intrinsically subjective nature of the phenomenon: what one reader finds funny another may not, which makes both the modelling and the evaluation of humor uniquely challenging. Figurative language, and humor in particular, has been a productive subject for shared tasks for several years[9, 10, 11].

SemEval-2015 Task 11 examined the impact of figurative devices such as metaphor and irony on sentiment analysis, and SemEval-2017 Task 6 asked systems to reproduce the ranking that a comedy programme's audience assigned to humorous tweets. The HAHA series, organised by Grupo PLN-InCo, has been a central reference point for humor in Spanish, with editions at IberEval 2018 [5], IberLEF 2019 [9], and IberLEF 2021 [11], all centred on humor detection and analysis. SemEval-2021 Task 7 [11] coupled humor detection with offence detection, and the HUHU task at IberLEF 2023 [12] focused specifically on identifying and classifying hurtful humor. More recently, interest has shifted toward computational humor *generation*, reflected in initiatives such as MWAHAHA, the SemEval 2026 Task 1 on humor generation, and the Arabic counterpart ARHAHA.

Against this backdrop, the 2026 edition of HAHA (Humor Analysis based on Human Annotation and Automatic Humor Generation), organised within IberLEF 2026 [13, 14], moves beyond detection alone and proposes three subtasks aimed at deepening our understanding of computational humor in Spanish. The first, *Humor Detection*, asks whether a news headline is a joke, distinguishing satirical headlines from genuine ones—a domain in which humorous and non-humorous content can be especially hard to tell apart. The second, *LLM-generated humor detection*, asks whether a joke inspired by a news headline was written by a human or produced by a large language model (LLM). The third, *Humor Generation*, requires systems to generate jokes from a news headline and is evaluated through human-preference judgments in LLM arena-style pairwise battles, with systems ranked on an Elo-based leaderboard. The two detection subtasks are scored by the F1 of the positive class (the *humorous* and the *automatic* class, respectively).

This paper describes the participation of the UC-UCO-Plenitas team in HAHA 2026. We present our systems for the subtasks we addressed, motivate the design choices behind them, and analyse our results in the context of the shared task. We pay particular attention to the twin challenges that this edition brings to the fore: the fine-grained, domain-specific discrimination required to tell satirical from real headlines, and the increasingly subtle boundary between human- and machine-authored humor as generative models improve. The remainder of the paper is organised as follows. Section 2 describes the task and the data; Section 3 details our methodology; Section 4 reports and discusses the results; and Section 5 concludes with directions for future work.

2. Methodology

2.1. Subtasks

The HAHA 2026 shared task at IberLEF 2026 proposes three subtasks aimed at deepening the computational understanding of humor in Spanish, and our system addresses all three within a single configurable pipeline. The first subtask [14], Humor Detection, is a binary classification of a news headline as satirical (a joke) or real; its main metric is the F1 score of the *humorous* class, and the difficulty lies in a domain where humorous and non-humorous content can be hard to tell apart. The second subtask, LLM-generated humor detection, is a binary classification of a headline-and-joke pair according to whether the joke was written by a human or generated by a large language model (LLM); its main metric is the F1 score of the *automatic* class. The third subtask, Humor Generation, requires the system to generate a joke from a news headline; it is evaluated through human-preference judgments in LLM arena-style pairwise battles and ranked on an Elo-based leaderboard. The first two subtasks are framed as discriminative problems and share a common ensemble architecture, whereas the third is a conditional generation problem and uses a retrieval-augmented generation pipeline.

2.2. Evaluation Protocol

We follow the official metrics of each subtask: the F1 of the *humorous* class for Subtask 1, the F1 of the *automatic* class for Subtask 2, and the Elo ranking derived from pairwise human preference battles for Subtask 3. Because the test phase provides no gold labels, the test-time pipeline only generates predictions and packages the submission archive; quantitative model selection is therefore performed during the development phase. For the two classification subtasks we tune the ensemble configuration — the embedding encoder, the embedding-classifier strategy, and the relative weight of the embedding and LLM components — on the labelled development data before committing to a test run. Submission integrity is verified programmatically prior to packaging: each output file is checked to contain exactly the expected columns (*id* plus the task-specific prediction column), the same number of rows as the evaluation file, and the identical *id* ordering, so that the predictions stay aligned with the official evaluation set.

2.3. Dataset

The data for all three subtasks are provided as tab-separated (TSV) files: a labelled trial split, a development split, and an unlabelled test split [14]. The trial split is the primary source of supervision; when the development split also carries gold labels, it is concatenated with the trial split to enlarge the training pool for the classifiers, otherwise only the trial split is used. For Subtask 1 the relevant fields are the headline and an optional context, which are concatenated into a single text; for Subtask 2 the headline and the associated joke are concatenated with a separator into a single text. For Subtask 3 we additionally build a retrieval pool of human-authored jokes by selecting, from the Subtask 2 trial (and from the development split when it contributes human-labelled jokes), the headline–joke pairs annotated as *human*, after de-duplication and removal of empty entries. All text is kept in its original Spanish form, and the pipeline preserves the per-row *id* so that predictions can be realigned with the evaluation files at submission time.

2.4. Sentence Embeddings

The text representation underlying the two classification subtasks and the Subtask 3 retrieval step is built on multilingual sentence embeddings. Each input text is encoded into a fixed-length dense vector with a pre-trained sentence-transformer; the encoder is selected at configuration time, with candidates including `intfloat/multilingual-e5-large-instruct` (the primary choice), `intfloat/multilingual-e5-large`, and `BAAI/bge-m3`. Embeddings are L2-normalised at encoding time so that cosine similarity reduces to an inner product, which benefits both the nearest-neighbour classifier and the retrieval step.

On top of these embeddings we apply a lightweight classifier that exposes class probabilities. Two interchangeable strategies are supported: a regularised logistic-regression head with balanced class weights, used when the training pool is large enough, and a weighted k-nearest-neighbour scheme that aggregates the cosine similarities of the k most similar training items into per-class scores, used when supervision is scarce. The same encoder serves the Subtask 3 retriever, where the evaluation headline is matched against the embedded pool of human jokes by cosine similarity to select the most relevant few-shot examples.

2.5. Large Language Models

In parallel with the embedding classifier, the pipeline incorporates a generative LLM served through Ollama, supporting both a local endpoint and a hosted cloud endpoint, with the model selected by configuration (for example a GLM- or Gemma-family cloud model locally backed by a smaller LLaMA model). For the two classification subtasks the LLM operates zero-shot under a strict system prompt that fixes its role and constrains it to answer with a single label word: *satirical* or *real* for Subtask 1, and *machine* or *human* for Subtask 2, with the latter prompt additionally cueing the model toward

stylistic discriminators of machine-generated text. Decoding is deterministic (temperature zero) for classification, a robust parser maps the single-word response back to the label inventory, and failed or ambiguous generations are handled by retries and a neutral fallback distribution. For Subtask 3 the LLM acts as a joke generator under a comedic system prompt, with a higher temperature to encourage creative and diverse output, and a post-processing step strips boilerplate prefixes and surrounding quotation marks from the generated joke. For example, the prompt instructed the model to generate a short original joke in Spanish inspired by the input headline, using the retrieved examples only as style references and avoiding direct copying.

2.6. System Architecture

The system is implemented as a single configurable pipeline, summarised in Figure 1, that routes each subtask through the appropriate strategy. For Subtasks 1 and 2 it uses a two-component ensemble. Component A encodes the input text with the multilingual sentence encoder and predicts class probabilities with the logistic-regression or k-nearest-neighbour head described above. Component B queries the zero-shot LLM classifier, which returns a calibrated probability over the same label set. The two probability vectors are then combined by a weighted average — with a larger weight on the LLM component than on the embedding component — and the final label is the arg-max of the combined distribution; if the LLM is unavailable, the pipeline falls back gracefully to the embedding component alone. For Subtask 3 the architecture is a retrieval-augmented generation pipeline: the evaluation headline is embedded and used to retrieve the most similar human-authored jokes from the indexed pool, those examples are inserted into a few-shot prompt as style references, and the LLM generates a new, original joke conditioned on the headline; when no retrieval pool is available the pipeline degrades to pure generation, and a deterministic placeholder is emitted only if generation fails entirely.

Finally, predictions for each subtask are written to separate TSV files — the predicted label in the *tag* column for the classification subtasks and the generated joke in the *text* column for the generation subtask — with the per-row *id* preserved. The output files are then validated for column structure, row count, and identifier ordering and compressed, without sub-directories, into the archive expected by the evaluation platform.

3. Results

Development Phase Results.

Table 1 reports the internal score of each component in isolation and of the LLM–embedding combinations. Two trends are clear. First, the hybrid configurations substantially outperform their isolated parts: standalone encoders score between 0.42 and 0.53 and standalone LLMs between 0.40 and 0.60, whereas three of the four LLM+embedding combinations exceed 0.80, providing direct empirical justification for the ensemble design. Second, the gemma4-cloud backbone is consistently the strongest, and—although it is unremarkable on its own (0.58)—it reaches the top score when paired with either E5 variant. Notably, the ranking of encoders reverses between the isolated and combined settings: `BAAI/bge-m3` is the best standalone encoder (0.53) yet the E5 variants, weakest in isolation, combine most effectively with gemma4. We caution that the two configurations reaching a perfect 1.00 should be interpreted carefully: such scores, obtained on the internal validation sample, are likely optimistic and may not transfer to the full test set; we therefore treat the relative ordering of methods, rather than the absolute values, as the reliable signal.

Table 2 reports the best submission per team on Task 1, ranked by F1. Our system, ymlopez (UCO-UC-CICESE-Plenitas), ranks first with an F1 of 0.9389, narrowly ahead of the second-placed namdang (0.9361)—a margin of only 0.0028—but clearly above the rest of the field, leading the third-ranked team (yemen2016, 0.8615) by more than 0.07 F1. The result is driven by the highest precision of the entire field (0.9685), combined with a strong recall (0.9111): our system is the most reliable when predicting the humorous class without sacrificing coverage. Interestingly, the two leading systems reach almost identical F1 from different precision/recall trade-offs—ours is slightly more precision-oriented

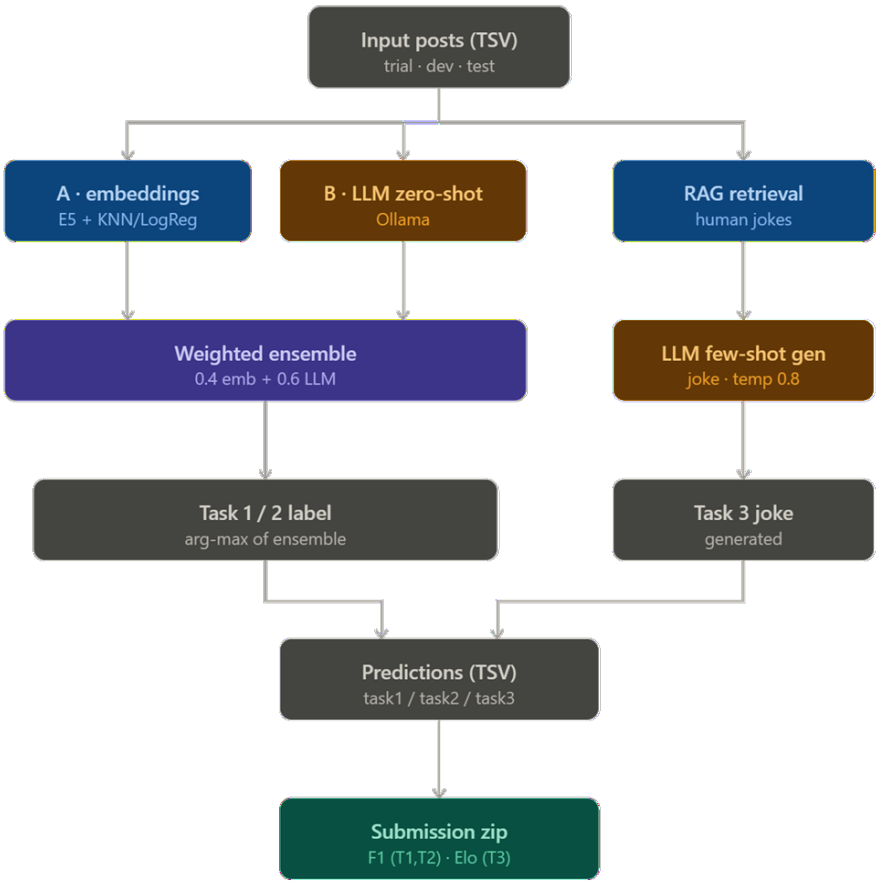

Figure 1: Overview of the proposed HaHa pipeline.

Table 1
Internal comparisons on the development phase. Scores correspond to the F1 measure obtained on the validation split used for model selection

Method	Score
gemma4-cloud+intfloat/multilingual-e5-large-instruct	1.00
glm-5.1:cloud+BAAI/bge-m3	0.81
gemma4-cloud+intfloat/multilingual-e5-large	1.00
gemma4-cloud+BAAI/bge-m3	0.83
BAAI/bge-m3	0.53
intfloat/multilingual-e5-large	0.42
multilingual-e5-large-instruct	0.51
oss-gpt120-cloud	0.40
gemma4-cloud	0.58
glm-5.1:cloud	0.60

(0.9685/0.9111) than namdang's (0.9504/0.9222)—indicating that top performance on this subtask is attainable through complementary balancing strategies. Overall, the field separates into three tiers: a tight top pair above 0.93, a middle group between 0.71 and 0.86, and a tail below 0.66; UCO-UC-CICESE-Plenitas leads the first.

Table 3 reports the best submission per team on Task 2, ranked by F1. Our system, ymlopez (UCO-UC-CICESE-Plenitas), ranks first with an F1 of 0.8329, ahead of the second-placed namdang (0.7760) by a clear 0.057 margin. Beyond leading in F1, our system is the only one with balanced precision and recall

Table 2
Task 1 (Development Phase) - Best submission per user (ranked by F1).

User	Submission Id	Precision	Recall	F1
ymlopez(UCO-UC-CICESE-Plenitas)	754570	0.9685	0.9111	**0.9389**
namdang	736367	0.9504	0.9222	0.9361
yemen2016	723726	0.8000	0.9333	0.8615
m_buch_135	745457	0.7450	0.8333	0.7867
matej2002	722721	0.6646	0.8000	0.7261
matej2002a	722362	0.7021	0.7333	0.7174
matej2002c	723712	0.6779	0.7481	0.7113
franrodrigo	695366	0.7673	0.5741	0.6568
edo2002	721551	0.6904	0.6111	0.6483
sebastian	733505	0.6199	0.6222	0.6211
ossey	757862	0.6274	0.6111	0.6191
joel	728863	0.5488	0.4370	0.4866

(0.8258 / 0.8400): every other team is strongly recall-biased at the expense of precision, an effect that is extreme for the most coverage-oriented systems (namdang 0.6512/0.9600; yemen2016 0.4971/0.9829). This indicates that the rest of the field tends to over-predict the positive class, whereas our system maintains reliability in both directions—precisely the behaviour rewarded by the official F1 metric of the positive class.

Our precision (0.8258) exceeds the next-best precision (joel, 0.7044) by more than 0.12, which is especially valuable in this subtask, where a false positive amounts to mislabelling human-written humor as machine-generated. Overall, UCO-UC-CICESE-Plenitas leads Task 2 as the only well-calibrated system in the field.

Table 3
Task 2 (Development Phase) - Best submission per user (ranked by F1).

User	Submission Id	Precision	Recall	F1
ymlopez(UCO-UC-CICESE-Plenitas)	754375	0.8258	0.8400	**0.8329**
namdang	707901	0.6512	0.9600	0.7760
joel	728863	0.7044	0.8171	0.7566
m_buch_135	754939	0.6275	0.7314	0.6755
yemen2016	718759	0.4971	0.9829	0.6603
franrodrigo	695375	0.5610	0.7886	0.6556

For Task 3, we submitted multiple configurations during the development phase to assess the impact of retrieval augmentation and generation settings. The explored variants differed mainly in the number of retrieved examples, the temperature used during generation, and the selected LLM backbone. In general, configurations without retrieval or with fewer retrieved examples tended to obtain lower ratings, whereas the best-performing run combined RAG-based few-shot prompting with a moderate temperature, which was subsequently selected for the evaluation phase.

Table 4 reports the best submission per team on Task 3 (humor generation), ranked by Elo rating from arena-style human-preference battles. Our system, ymlopez (UCO-UC-CICESE-Plenitas), obtains the highest point rating (1467), ahead of franrodrigo (1371) and yemen2016 (1268). However, all three teams share Rank 1: the official ranking treats them as statistically tied at the top, because their 95% confidence intervals overlap substantially (ours [1322, 1654] overlaps both [1255, 1565] and [1202, 1355]). This overlap means the rating differences are not statistically conclusive. The interpretation is further nuanced by sample size: our rating and franrodrigo's rest on only 55 and 53 battles, yielding wide intervals of roughly ± 150 points, whereas yemen2016 accumulates 345 votes and a much tighter interval (± 75). Our system therefore achieves the strongest point estimate of generation quality, but

within a shared first place where the leading ratings remain uncertain; a larger number of battles would be required to establish a statistically significant ranking.

Table 4
Task 3 (Development Phase) - Best submission per user (ranked by Elo rating).

User	Submission ID	Rank	Rating	95% CI	Votes
ymlopez(UCO-UC-CICESE-Plenitas)	754375	1	1467	[1322, 1654]	55
franrodrigo	695371	1	1371	[1255, 1565]	53
yemen2016	726740	1	1268	[1202, 1355]	345

Evaluation Phase Results

Table 5 isolates the effect of the LLM backbone by fixing the embedding encoder to `intfloat/multilingual-e5-large-instruct` across all three configurations. The `glm-5.1-cloud` backbone yields the best score (0.95), closely followed by `gemma4-cloud` (0.93), while oss-gpt120-cloud trails by more than 0.10 (0.8489). Two observations follow. First, the two top backbones are effectively tied (a 0.02 gap), indicating that either is a sound choice when paired with this encoder. Second, the larger `oss-gpt120` model is the weakest of the three, reinforcing the trend observed throughout our experiments that backbone family and task alignment matter more than raw parameter count for this task. We therefore adopt `glm-5.1-cloud` as the backbone for our final configuration.

Table 5
Internal comparisons on the evaluation-phase model-selection experiments. Scores correspond to F1 on the development data using the same validation protocol

Submission	Score
oss-gpt120-cloud+intfloat/multilingual-e5-large-instruct	0.8489
glm-5.1:cloud+intfloat/multilingual-e5-large-instruct	0.95
gemma4-cloud+intfloat/multilingual-e5-large-instruct	0.93

Table 6 reports the best submission per team on Task 1, ranked by F1. Our system, ymlopez (UCO-UC-CICESE-Plenitas), ranks second of eight with an F1 of 0.9525 (accuracy 0.9533), behind michaelibrahim (0.9933) and ahead of namdang_2006 (0.9381) and the rest of the field. Our result combines the second-highest precision in the field (0.9690) with a strong recall (0.9367), and it clearly outperforms the official baseline by more than 0.33 F1 (0.6252). While the leading system achieved an exceptionally high score, our approach maintains a balanced precision–recall trade-off, resulting in a robust and reliable classifier that performs consistently across both classes.

Table 6
Task 1 (Evaluation Phase) - Best submission per user (ranked by F1).

User	Submission ID	Precision	Recall	F1	Accuracy
michaelibrahim	776365	0.9933	0.9933	0.9933	0.9933
ymlopez(UCO-UC-CICESE-Plenitas)	762070	0.9690	0.9367	0.9525	0.9533
namdang_2006	758972	0.9681	0.9100	0.9381	0.9400
yemen2016	773525	0.8201	0.8967	0.8567	0.8500
m_buch_135	762280	0.7872	0.8633	0.8235	0.8150
baseline	778282	0.6923	0.5700	0.6252	0.6583
edo2002	769791	0.7225	0.5033	0.5933	0.6550
arjun108	759419	0.8630	0.4200	0.5650	0.6767

Table 7 reports the best submission per team on Task 2, ranked by F1. Here our system, ymlopez (UCO-UC-CICESE-Plenitas), ranks sixth of seven with an F1 of 0.7333 (accuracy 0.7382), ahead only of

the baseline (0.6597) and 0.143 below the leader (michaelibrahim, 0.8764). Unlike Task 1, where our system led on the strength of its precision, on Task 2 it records the lowest precision of the field (0.7472) together with the lowest recall (0.7200); the limitation is one of absolute discriminative power on both axes rather than of precision/recall balance. The contrast between subtasks is the most informative finding: the same system that ranks at the top on Task 1 (satirical-vs-real headline detection) drops to sixth on Task 2 (human-vs-LLM humor detection), a gap of roughly 0.22 F1. This localises the weakness of our approach precisely—it discriminates satirical from real headlines reliably but struggles to separate human-written from machine-generated humor, a distinction that is intrinsically subtler and on which the lexical and semantic cues effective in Task 1 appear not to transfer.

Table 7
Task 2 (Evaluation Phase) - Best submission per user (ranked by F1).

User	Submission ID	Precision	Recall	F1	Accuracy
michaelibrahim	776619	0.8764	0.8764	0.8764	0.8764
yemen2016	773641	0.8274	0.9236	0.8729	0.8655
m_buch_135	771442	0.8577	0.8327	0.8450	0.8473
namdang_2006	769240	0.8339	0.8400	0.8370	0.8364
arjun108	759419	0.8605	0.8073	0.8330	0.8382
ymlopez(UCO-UC-CICESE-Plenitas)	758939	0.7472	0.7200	0.7333	0.7382
baseline	778282	0.5165	0.9127	0.6597	0.5291

Table 8 reports the final standings for Task 3, where systems are ranked by a rating derived from pairwise comparisons over a comparable number of votes (344–347 per system). Our submission (`ymlopez`, UCO-UC-CICESE-Plenitas) achieved the highest rating of the evaluation, 1136 with a 95% confidence interval of [1108, 1163], placing it first. The second-ranked system (`michaelibrahim`) obtained a rating of 1110 [1083, 1138]. Because the confidence intervals of the two leading systems overlap substantially (over the range [1108, 1138]), the difference between them is not statistically significant, and both are therefore assigned the top rank. This indicates that the two approaches are statistically indistinguishable at the head of the leaderboard rather than separated by a meaningful margin. A clear and statistically significant gap separates the two leading systems from the remainder of the field. The third-ranked system (`yemen2016`) scored 919 [893, 944], whose interval lies entirely below those of the top two, confirming a genuine performance difference rather than sampling noise. The official `baseline` ranked last at 835 [810, 873]. Our system thus improves over the baseline by 301 rating points, with non-overlapping confidence intervals throughout, providing strong evidence that the gain is robust and not attributable to evaluation variance. The comparable vote counts and the similar interval widths across all systems (approximately ± 27 points) suggest that the ratings are estimated with consistent precision, lending further confidence to these conclusions.

Table 8
Task 3 (Evaluation Phase) - Best submission per user.

User	Submission ID	Rank	Rating	95% CI	Votes
ymlopez(UCO-UC-CICESE-Plenitas)	758939	1	**1136**	[1108, 1163]	344
michaelibrahim	776934	1	1110	[1083, 1138]	344
yemen2016	773641	3	919	[893, 944]	345
baseline	778282	4	835	[810, 873]	347

4. Conclusions

We have presented a unified approach to the three HAHA 2026 subtasks that combines dense multilingual representations with locally served large language models within a single configurable pipeline.

Our experiments confirm the value of the hybrid design: configurations that combine an encoder with an LLM substantially outperform their isolated components, providing direct empirical justification for the ensemble architecture. We also observe consistently that model family and task alignment matter more than raw parameter count, as illustrated by the fact that the largest model (`oss-gpt120-cloud`) is the weakest of those evaluated; on this basis we adopt `glm-5.1-cloud` together with `multilingual-e5-large-instruct` in our final configuration.

The evaluation-phase results reveal a markedly asymmetric profile. On Subtask 1 (satirical-vs-real headline detection) our system ranks second of eight, with an F1 of 0.9525 and a balanced precision/recall profile that exceeds the official baseline by a wide margin. On Subtask 2 (human-vs-LLM humor detection), by contrast, it drops to sixth of seven with an F1 of 0.7333, a fall of roughly 0.22 points relative to Subtask 1. This gap localises the limitation of our method precisely: the lexical and semantic cues that are effective for identifying satirical humor do not transfer to the subtler task of detecting automatic authorship. On the generation subtask, our system attains the highest Elo point estimate, although within a shared first place whose confidence intervals overlap, so that a larger number of battles would be required to establish a statistically conclusive ranking. Moreover, in Task 3, our system (`ymlopez`, UCO-UC-CICESE-Plenitas) obtained the highest rating of the evaluation (1136), securing first place and statistically tying with the next system, whose confidence interval it overlaps. Both leading systems clearly separated from the rest of the field, outperforming the official baseline by a substantial and statistically significant margin of 301 rating points with non-overlapping confidence intervals. These results confirm the effectiveness of our approach for Task 3 and establish it among the top-performing systems in the shared task, while also pointing to the close competition at the top as a promising direction for further refinement in future work.

As directions for future work, we plan to enrich the automatic-text-detection component with more specific stylistic and statistical discriminators—rather than relying on the same features that are useful for Subtask 1—to explore supervised fine-tuning of the LLMs instead of their purely zero-shot use, and to strengthen the evaluation of generation with a volume of pairwise comparisons sufficient to reduce the uncertainty of the Elo estimates.

Declaration on Generative AI

During the preparation of this work, the authors used Claude (Opus 4.8) and Grammarly for language revision, grammar and spelling correction, and assistance in the preparation of figures.

References

[1] S. Attardo, V. Raskin, Script theory revis(it)ed: Joke similarity and joke representation model, Humor: International Journal of Humor Research 4 (1991) 293–347. doi:10.1515/humr.1991.4.3-4.293.

[2] S. Attardo, C. F. Hempelmann, S. Di Maio, Script oppositions and logical mechanisms: Modeling incongruities and their resolutions, Humor 15 (2002) 3–46. doi:10.1515/humr.2002.004.

[3] S. Castro, M. Cubero, D. Garat, G. Moncecchi, Is this a joke? detecting humor in Spanish tweets, in: Ibero-American Conference on Artificial Intelligence (IBERAMIA), Springer International Publishing, 2016, pp. 139–150. doi:10.1007/978-3-319-47955-2_12.

[4] S. Castro, L. Chiruzzo, A. Rosá, D. Garat, G. Moncecchi, A crowd-annotated Spanish corpus for humor analysis, in: Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media, 2018, pp. 7–11. doi:10.18653/v1/W18-3502.

[5] S. Castro, L. Chiruzzo, A. Rosá, Overview of the HAHA task: Humor analysis based on human annotation at IberEval 2018, in: Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018), CEUR Workshop Proceedings, CEUR-WS.org, 2018.

[6] J. L. Fleiss, Measuring nominal scale agreement among many raters, Psychological Bulletin 76 (1971) 378–382. doi:10.1037/h0031619.
[7] J. Sjöbergh, K. Araki, Recognizing humor without recognizing meaning, in: Applications of Fuzzy Sets Theory (WILF 2007), Springer, 2007, pp. 469–476. doi:10.1007/978-3-540-73400-0_59.
[8] R. Mihalcea, C. Strapparava, 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 '05), Association for Computational Linguistics, Vancouver, British Columbia, Canada, 2005, pp. 531–538. doi:10.3115/1220575.1220642.
[9] L. Chiruzzo, S. Castro, M. Etcheverry, D. Garat, J. J. Prada, A. Rosá, Overview of HAHA at IberLEF 2019: Humor analysis based on human annotation, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019), CEUR Workshop Proceedings, CEUR-WS.org, Bilbao, Spain, 2019.
[10] L. Chiruzzo, S. Castro, A. Rosá, HAHA 2019 dataset: A corpus for humor analysis in Spanish, in: Proceedings of the 12th Language Resources and Evaluation Conference (LREC), 2020, pp. 5106–5112.
[11] L. Chiruzzo, S. Castro, S. Góngora, A. Rosá, J. A. Meaney, R. Mihalcea, Overview of HAHA at IberLEF 2021: Detecting, rating and analyzing humor in Spanish, Procesamiento del Lenguaje Natural 67 (2021) 257–268.
[12] R. Labadie-Tamayo, B. Chulvi, P. Rosso, Everybody hurts, sometimes. overview of HUrtful HUmour at IberLEF 2023: Detection of humour spreading prejudice in Twitter, Procesamiento del Lenguaje Natural (SEPLN) (2023).
[13] A. Bonet-Jover, J. Á. González-Barba, L. Chiruzzo, Overview of IberLEF 2026: Natural language processing challenges for spanish and other iberian languages, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2026), co-located with the 42nd Conference of the Spanish Society for Natural Language Processing (SEPLN 2026), CEUR Workshop Proceedings, CEUR-WS.org, 2026.
[14] L. Chiruzzo, I. Sastre, G. Rey, A. Martínez, A. Rosá, S. Góngora, S. Castro, V. Amoroso, J. Prada, J. Conde, G. Moncecchi, Overview of HAHA at IberLEF 2026: Humor analysis based on human annotation and automatic humor generation, Procesamiento del Lenguaje Natural 77 (2026).

A. Online Resources

The results of the HaHa 2026 are available via:

- HaHa - Humor Analysis based on Human Annotation and Automatic Humor Generation
- HaHa 2026 Code
- HaHa 2026 Development Phase Results
- HaHa 2026 Evaluation Phase Results
- HaHa 2026 Results