MNN at HAHA@IberLEF 2026 Task-Adaptive Reasoning and Fine-Tuning for Spanish Humor Detection

Dang Quoc Nam[1,2,*], Tran Trong Nhan[1,2], Truong Duc Manh[1,2] and Dang Van Thin[1,2]

[1]*University of Information Technology, Vietnam National University Ho Chi Minh City, Quarter 34, Linh Xuan Ward, Ho Chi Minh City, Vietnam*

[2]*Vietnam National University Ho Chi Minh City, Ho Chi Minh City, Vietnam*

Abstract

This paper describes our systems for the HAHA 2026 shared task at IberLEF 2026, addressing Subtask 1 (satirical news headline detection) and Subtask 2 (LLM-generated humor detection). We implement three approaches: a few-shot chain-of-thought-style prompting system that queries *gemma-4-31B-it*; an improved prompting variant with explicit linguistic criteria, diversity-based example selection, and self-consistency voting for Subtask 2; and a supervised fine-tuning pipeline based on BETO and XLM-RoBERTa encoders with five-fold stratified ensembling. The prompting-based systems obtained the strongest result on Subtask 1, with an official test F1 of 0.9381 for the positive (`satirical`) class. In contrast, supervised fine-tuning obtained the strongest result on Subtask 2, with an official test F1 of 0.8369 for the positive (`machine`) class. These results show that the two subtasks behave differently in practice: satirical headline detection is better handled by prompting a large instruction-following model, while LLM-generated joke detection benefits more from supervised learning over dataset-specific stylistic and length-related cues.

Keywords

Spanish humor detection, satirical news, LLM-generated text detection, few-shot prompting, chain-of-thought, BERT and XLM-RoBERTa fine-tuning, self-consistency voting, HAHA 2026

1. Introduction

Computational humor remains a challenging problem because humor is highly subjective, culturally situated, and expressed through pragmatic mismatch, irony, exaggeration, and script opposition. The HAHA shared-task series has contributed to the study of Spanish humor through benchmark datasets and evaluation settings for humor detection and analysis over several editions [1, 2]. The 2026 edition extends this line of work toward both satirical news detection and automatic humor generation analysis.

HAHA 2026 includes three subtasks. This paper addresses the first two. Subtask 1 asks systems to distinguish satirical news headlines from real news headlines. Subtask 2 asks systems to determine whether a joke inspired by a news headline was written by a human or generated by a large language model (LLM). Both subtasks are binary classification problems evaluated with the F1 score of the positive class: `satirical` for Subtask 1 and `machine` for Subtask 2.

The two subtasks differ not only in their input format but also in the type of evidence that may be useful for classification. In Subtask 1, the system must judge whether a news headline is plausible as real news or whether it contains satirical cues such as exaggeration, irony, or pragmatic incongruity. This makes semantic interpretation and background knowledge important. In Subtask 2, the system compares human-written jokes with LLM-generated jokes, where the released data shows clear stylistic and length-related differences. These observations motivated us to evaluate both prompting-based and supervised fine-tuning approaches rather than relying on a single modeling strategy.

We implement and compare three systems. System 1 uses few-shot prompting with chain-of-thought-style reasoning [3, 4] through an API that hosts `gemma-4-31B-it` [5]. System 2 modifies the prompting approach for Subtask 2 by using a more explicit linguistic prompt, diversity-based example selection,

IberLEF 2026, September 2026, León, Spain

*Corresponding author.

24521098@gm.uit.edu.vn (D. Q. Nam); 24521243@gm.uit.edu.vn (T. T. Nhan); 24521046@gm.uit.edu.vn (T. D. Manh); thindv@uit.edu.vn (D. V. Thin)

Table 1
Dataset sizes and label distributions. Test sets contain no gold labels.

Task	Split	Samples	Labels	Distribution
Subtask 1	Trial	24	yes	12 real / 12 satirical
	Dev-gold	540	yes	270 real / 270 satirical
	Trial+dev-gold	564	yes	282 real / 282 satirical
	Test	600	no	–
Subtask 2	Trial	12	yes	6 machine / 6 human
	Dev-gold	350	yes	175 machine / 175 human
	Trial+dev-gold	362	yes	181 machine / 181 human
	Test	550	no	–

and self-consistency voting. System 3 fine-tunes BETO for Subtask 1 and XLM-RoBERTa for Subtask 2 with five-fold stratified ensembling; both models are based on the Transformer encoder architecture [6]. We report the official test scores for all three systems and analyze how their behavior differs across the two subtasks.

2. Data Analysis

2.1. Dataset Overview

The HAHA 2026 organizers provided datasets comprising trial, development-gold, and unlabelled test splits for both subtasks. For Subtask 1, each instance contains a news headline, a short context, and metadata including the date and the source country (spanning six Spanish-speaking regions: Colombia, Mexico, Puerto Rico, Spain, Uruguay, and Venezuela). The target label is either `satirical` (the positive class) or `real`. For Subtask 2, each instance contains a news headline and a joke inspired by that headline, with the target label being `machine` (the positive class) or `human`.

Table 1 summarizes the dataset sizes and label distributions. In the released labeled splits, both subtasks are balanced with respect to their binary labels. To maximize the learning signal during supervised fine-tuning, we concatenate the trial and development-gold sets before producing the final test predictions.

2.2. Exploratory Analysis

Subtask 1 length. Satirical headlines tend to be slightly longer than real ones. In the combined labeled data, real headlines have a mean length of approximately 11.39 words and satirical headlines have a mean of 14.23 words, with medians of 11 and 14 respectively. Context length is comparable across labels (approximately 16 words for both), indicating that context acts as a supporting signal rather than a primary discriminator.

Subtask 2 length. A highly salient finding emerges in the context of Subtask 2. Human jokes have a mean length of about 13.06 words, while machine jokes average approximately 25.36 words—roughly twice as long (Table 2). Moreover, 84.5% of machine jokes contain at least 18 words, compared with only 21.5% of human jokes. We therefore treat length as an informative auxiliary cue rather than as a standalone decision rule, since a non-negligible fraction of human jokes also exceeds this threshold and some machine jokes are relatively short. This profound distributional difference motivates the explicit length hint used in System 3 (Section 3.3).

Beyond length, machine jokes in the data are more likely to exhibit a clean setup–transition–punchline structure, neutral vocabulary, and conventional punctuation. In contrast, human jokes are often shorter, more conversational, regionally marked, and structurally irregular.

Table 2
Word number statistics in the combined labeled data.

Task	Label	Mean	Median	Min	Max
Subtask 1 headline	real	11.39	11	2	24
	satirical	14.23	14	5	37
Subtask 2 joke	human	13.06	12	1	43
	machine	25.36	26	12	51

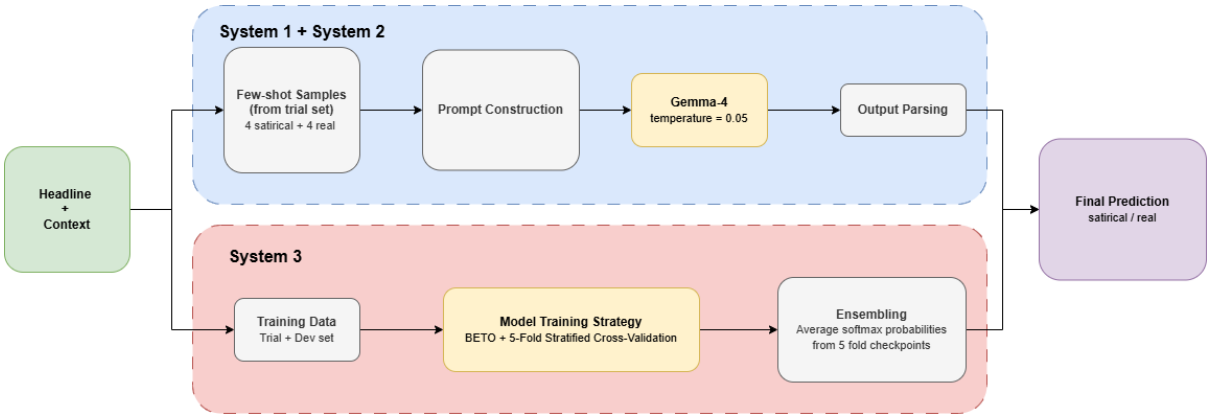

Figure 1: Pipeline comparison for Subtask 1. Systems 1 and 2 share the same few-shot prompting pipeline, while System 3 uses supervised fine-tuning with five-fold probability ensembling.

3. Methodology

3.1. Overview

This section describes the three implemented systems and the evaluation protocol used to compare them. The systems are organized into two methodological families: prompting-based inference with an instruction-following language model and supervised fine-tuning of BERT-based encoders. The goal is to compare these approaches under the same shared-task setting and to examine how they behave on the two different subtasks. The labeled trial and development-gold splits are used for prompt construction, exploratory analysis, supervised training, and fold-level validation. The held-out test labels are not used during training, prompt construction, validation, or model selection. Tables 3 and 4 summarize the key properties of each system for each subtask.

3.2. Subtask 1: Satirical News Headline Detection

Subtask 1 requires distinguishing satirical news headlines from real news headlines. We compare two approaches for this subtask: (1) prompting-based inference with a large instruction-following language model, and (2) supervised fine-tuning of a Spanish BERT encoder. For Subtask 1, Systems 1 and 2 are identical and therefore produce the same predictions. The distinction between Systems 1 and 2 applies only to Subtask 2, where System 2 introduces an improved prompting strategy. System 3, in contrast, uses supervised fine-tuning and five-fold probability ensembling.

Figure 1 illustrates the two processing branches for Subtask 1. The prompting branch uses the headline and context to construct a few-shot reasoning prompt, queries the LLM, and parses the final label. The fine-tuning branch trains a Spanish transformer classifier on the combined trial and development-gold data and averages predictions from five fold checkpoints.

Prompting-based systems. For Systems 1 and 2, Subtask 1 is handled through few-shot prompting with explicit reasoning instructions. The model used for inference is `gemma-4-31B-it`. For each input

Table 3
Comparison of the implemented systems for Subtask 1.

System	Method	Input construction	Prediction strategy
System 1 and 2	Few-shot prompting	Headline, context, and trial-set demonstrations	Single API call and label parsing
System 3	Supervised fine-tuning	Headline concatenated with context	Five-fold probability ensemble

instance, the system builds a prompt using four `satirical` and four `real` demonstrations from the trial split. Each demonstration contains the headline, a short prefix of the context, a brief reasoning statement, and the gold label.

The system prompt frames the model as an expert in Spanish-language media analysis. It asks the model to identify signals of satire, such as absurd exaggeration, irony, humor, impossible situations, or a departure from neutral journalistic style. It also asks the model to consider signals of real news, such as factual plausibility, neutral wording, and standard journalistic tone. The model is instructed to end its answer with a constrained label line in the following format:

```
ETIQUETA: <satirical|real>
```

After each API call, a parser searches the response for the `ETIQUETA:` marker and extracts one of the valid labels. If the marker is missing, the parser falls back to searching for a valid label anywhere in the response. If the API call fails after up to three retries, each with a 90-second timeout, or if no valid label can be parsed, the system assigns the default positive label, `satirical`. This guarantees a complete submission, although it may introduce a small positive-class bias in rare failure cases.

Supervised fine-tuning system. System 3 treats Subtask 1 as a supervised binary classification problem. The trial and development-gold splits are concatenated to form the labeled training pool. We fine-tune `dccuchile/bert-base-spanish-wwm-cased` (BETO), a Spanish BERT model trained with whole-word masking [7, 8]. The input sequence is constructed by concatenating the `headline` and `context` fields. If the context field is missing or empty, the system uses only the headline.

The system uses five-fold stratified cross-validation with `random_state=42`. In each fold, the model is trained on four folds and validated on the remaining fold. The best checkpoint for each fold is selected according to validation F1 on the positive class, `satirical`. At test time, the five checkpoints produce softmax probabilities for each instance. The probabilities are averaged, and the label with the highest averaged probability is selected as the final prediction.

3.3. Subtask 2: LLM-generated Humor Detection

Subtask 2 asks whether a joke inspired by a news headline was generated by a language model or written by a human. In contrast to Subtask 1, the three systems follow different strategies for this task. System 1 uses the initial few-shot prompting pipeline. System 2 strengthens the prompting pipeline through diversity-based demonstration selection, explicit linguistic criteria, and self-consistency voting. System 3 uses supervised fine-tuning with an explicit length marker motivated by the exploratory analysis. Figure 2 illustrates the three processing branches for Subtask 2.

System 1. System 1 uses the same general prompting framework as in Subtask 1. The prompt contains three `machine` and three `human` demonstrations from the trial split. Each demonstration includes the headline, the joke, a brief reasoning statement, and the gold label. The prompt asks the model to detect generic structure, predictable setup–punchline patterns, and neutral wording as possible signs of machine generation, while treating regional references, unexpected turns, wordplay, and natural imperfections as possible signs of human authorship. The model output is constrained to:

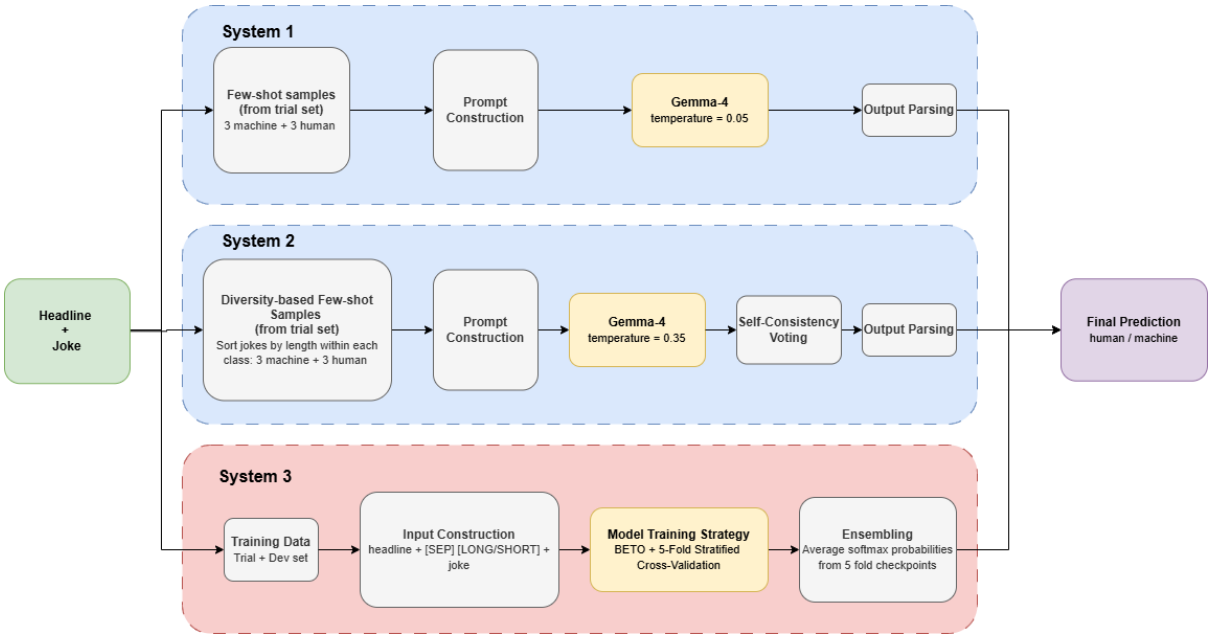

Figure 2: Pipeline comparison for Subtask 2. The three systems use different strategies: basic few-shot prompting, improved prompting with self-consistency voting, and supervised fine-tuning with an explicit length marker.

`ETIQUETA: <machine|human>`

Inference uses temperature 0.05. The parser extracts the final label from the `ETIQUETA:` line. If the API call fails after up to three retries, each with a 90-second timeout, or if no valid label can be parsed, the system assigns the default positive label, `machine`.

System 2. System 2 only modifies the Subtask 2 prompting pipeline. First, instead of taking the first examples from each class, it sorts jokes within each label by character length and selects examples at evenly spaced positions. This produces a fixed length-diverse demonstration set that is reused for all input instances; the demonstrations are not re-sampled for each test instance. Second, it uses a more explicit linguistic prompt: machine-generated jokes are described as more polished, structurally regular, topically direct, and conventionally punctuated, whereas human-written jokes are described as more conversational, irregular, locally marked, or unexpectedly absurd. The prompt also warns the model not to equate joke quality with human authorship.

Finally, System 2 applies self-consistency voting. For each instance, the model is queried three times with temperature 0.35. The final label is selected by majority vote [9], and the vote ratio of the winning label is recorded as a confidence score.

Prompt format. For both subtasks, the user prompt contains labeled demonstrations followed by the target input. Subtask 1 demonstrations use the format `Titular`, optional `Contexto`, a short reasoning statement, and the final label. Subtask 2 demonstrations use `Titular`, `Chiste`, a short reasoning statement, and the final label. In all prompting systems, the required final output is restricted to `ETIQUETA: <label>`.

System 3. System 3 fine-tunes `xlm-roberta-base` [10] on the combined trial and development-gold data. The input sequence is:

`headline [SEP]` *length_hint* `joke`

Table 4
Comparison of the implemented systems for Subtask 2.

System	Method	Input or prompt construction	Prediction strategy
System 1	Basic few-shot prompting	Headline, joke, and trial demonstrations from each class	Single API call and label parsing
System 2	Improved prompting	Diverse demonstrations and explicit linguistic criteria	Three API calls with majority vote
System 3	Supervised fine-tuning	Headline, length marker, and joke	Five-fold probability ensemble

Table 5
Fine-tuning hyperparameters for System 3.

Hyperparameter	Subtask 1	Subtask 2
Base model	BETO cased	XLM-RoBERTa base
Input fields	headline + context	headline + length marker + joke
Maximum length	128 tokens	128 tokens
Batch size	16	8
Epochs	5	6
Learning rate	2×10^{-5}	2×10^{-5}
Weight decay	0.01	0.01
Warmup ratio	0.1	0.1
Dropout	0.1	0.1
Number of folds	5	5
Gradient checkpointing	no	yes
Checkpoint criterion	positive-class F1	positive-class F1

Table 6
Official test F1 scores for the implemented systems.

System	Subtask 1 F1	Subtask 2 F1
System 1	**0.9381**	0.6039
System 2	**0.9381**	0.7018
System 3	0.8294	**0.8369**

where *length_hint* is `[LONG]` if the joke contains at least 18 words and `[SHORT]` otherwise. This marker encodes the strong length difference observed in Section 2.2, while still allowing the model to learn lexical, stylistic, and headline–joke interaction patterns from the full input. As in Subtask 1, System 3 uses five-fold stratified validation, selects the best checkpoint by positive-class F1, and averages the five fold probability vectors for test prediction.

Training configuration for System 3. Table 5 summarizes the fine-tuning configuration. Both subtasks use `AutoModelForSequenceClassification` with `num_labels=2`, AdamW, cross-entropy loss, a linear learning-rate schedule with warmup, weight decay, and gradient clipping. Inputs are tokenized with truncation and fixed-length padding to 128 tokens. For Subtask 2, the batch size is reduced to 8 and gradient checkpointing is enabled to reduce GPU memory usage.

4. Results

Table 6 reports the official test F1 scores for the three implemented systems. The results show different behavior across the two subtasks: prompting-based systems perform best on Subtask 1, whereas

Table 7
Official leaderboard context after deduplicating repeated submissions with the same team name and identical F1 score.

Task	Team / System	Rank	F1
Subtask 1	michaelibrahim	1	0.9933
	ymlopez	2	0.9525
	System 1 (ours)	**3**	**0.9381**
	System 2 (ours)	**3**	**0.9381**
	System 3 (ours)	8	0.8294
	baseline	11	0.6252
Subtask 2	michaelibrahim	1	0.8764
	yemen2016	2	0.8729
	System 3 (ours)	**5**	**0.8369**
	System 2 (ours)	10	0.7018
	baseline	11	0.6597
	System 1 (ours)	14	0.6039

supervised fine-tuning performs best on Subtask 2.

Table 7 places these results in the official leaderboard context. To avoid counting repeated submissions as separate ranks, entries with the same team name and identical F1 score are collapsed. Under this deduplicated view, Systems 1 and 2 are tied for rank 3 on Subtask 1, while System 3 reaches rank 5 on Subtask 2.

The leaderboard confirms that the relative behavior of the systems is not merely an internal comparison among our own submissions. For Subtask 1, the prompting-based systems are competitive with the strongest submissions and clearly outperform the official baseline. For Subtask 2, the supervised fine-tuning system is our strongest submission and is close to the top-ranked systems, whereas the initial prompting system falls below the baseline.

4.1. Subtask 1 Results

For Subtask 1, Systems 1 and 2 obtained the best official test F1 score among our submissions, both reaching 0.9381 and rank 3 in the deduplicated leaderboard. This tie is expected because System 2 keeps the same Subtask 1 pipeline as System 1. Both systems rely on few-shot prompting with an instruction-following language model, using headline and context information together with trial-set demonstrations. This strategy appears well suited to satirical headline detection, where the model must recognize implausible events, ironic framing, and pragmatic incongruity rather than only surface lexical patterns.

System 3 obtained a lower F1 score of 0.8294 and ranks lower than the prompting-based systems. Although supervised fine-tuning can learn task-specific lexical and contextual patterns, the labeled training pool is relatively small. As a result, the fine-tuned classifier may not capture enough background knowledge or semantic plausibility judgments to distinguish subtle satire from unusual but real news. This explains why the prompting systems, which can rely on the broader prior knowledge of a large instruction-following model, are more effective for this subtask.

4.2. Subtask 2 Results

For Subtask 2, the trend is reversed. System 1 achieves an official test F1 of 0.6039 and ranks below the baseline. This suggests that the initial prompting strategy is too conservative in assigning the positive class. On the official leaderboard, System 1 has high precision but much lower recall, meaning that when it predicts `machine`, it is often correct, but it misses many machine-generated jokes. Since the official metric is F1 for the `machine` class, this low recall strongly reduces the final score.

System 2 improves the Subtask 2 F1 score from 0.6039 to 0.7018 and surpasses the baseline. This shows that the modifications introduced in System 2 are useful: length-diverse demonstrations expose the model to a broader range of examples, the explicit linguistic prompt gives clearer decision criteria, and self-consistency voting reduces some single-call instability. However, System 2 still remains far below System 3, suggesting that prompting alone does not fully capture the dataset-specific regularities needed for this task.

System 3 achieves the best Subtask 2 result among our submissions, with an official test F1 of 0.8369 and rank 5 in the deduplicated leaderboard. This supports the exploratory analysis: LLM-generated jokes in the dataset tend to be longer, more structurally polished, and more regular than human-written jokes. By fine-tuning directly on the labeled data and adding an explicit length marker, System 3 can exploit these stylistic and distributional cues more consistently than the prompting-based systems.

Overall, the comparison shows that the best-performing approach depends on the subtask. For Subtask 1, prompting is more effective, likely because the task requires semantic interpretation, background knowledge, and recognition of satirical incongruity. For Subtask 2, supervised fine-tuning is more effective, as the labeled data contains clearer stylistic and length-related regularities. This practical difference explains why the strongest configuration uses prompting for Subtask 1 and supervised fine-tuning for Subtask 2.

5. Discussion

The results reveal clear task-dependent behavior, but the proposed systems still have several limitations. The prompting-based systems depend on an LLM, which makes the results sensitive to prompt wording, demonstration selection, decoding behavior, and possible backend changes. Although self-consistency voting improves System 2 on Subtask 2, it also increases inference cost and cannot fully correct systematic errors when the model follows the same incorrect reasoning pattern across multiple calls.

The supervised fine-tuning system has a different limitation. In Subtask 2, the explicit length marker is motivated by the strong length difference observed in the labeled data: machine jokes are substantially longer on average than human jokes, and a much larger proportion of machine jokes exceed the 18-word threshold (Table 2). However, this cue may not generalize if future LLM-generated jokes become shorter or if human-written jokes become more polished and structurally complete. Therefore, the fine-tuned model may partly rely on dataset-specific surface regularities rather than deeper authorship or humor signals.

A natural direction for improvement is to combine the strengths of prompting and fine-tuning more explicitly. For example, future systems could use calibrated confidence scores or out-of-fold predictions to decide which model to trust for each instance. Further ablation studies are also needed to measure the separate contribution of prompt design, self-consistency voting, and the length marker. Finally, a more detailed error analysis by country, topic, joke length, regional vocabulary, and humor type would help identify when each modeling strategy succeeds or fails.

6. Conclusion

This paper presented three systems for HAHA 2026 Subtasks 1 and 2, comparing prompting-based inference with supervised BERT-based fine-tuning. The official test results show that the two subtasks are better handled by different approaches. For Subtask 1, the prompting-based systems achieved the best F1 score of 0.9381, indicating that a large instruction-following model is effective for recognizing satirical cues in news headlines. For Subtask 2, the supervised fine-tuning system achieved the best F1 score of 0.8369, showing that direct learning from labeled data is useful for detecting LLM-generated jokes. The main lesson from the experiments is practical: a single modeling strategy does not work equally well for both subtasks. Prompting is useful when the task requires broader semantic interpretation and background knowledge, while fine-tuning is useful when the dataset contains stable stylistic or

distributional patterns. This motivates a task-adaptive workflow in which the choice of method is guided by the characteristics of each subtask rather than by a fixed preference for prompting or fine-tuning.

Declaration on Generative AI

Generative AI tools were used only to support language refinement and improve the clarity of the writing. All methodological decisions, data analysis, experimental interpretation, and conclusions were made by the authors. The authors take full responsibility for the content and integrity of this manuscript.

References

[1] 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).

[2] 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-WS. org, 2026.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[4] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., Chain-of-thought prompting elicits reasoning in large language models, Advances in neural information processing systems 35 (2022) 24824–24837.

[5] Google DeepMind, Gemma 4 model card, https://ai.google.dev/gemma/docs/core/model_card_4, 2026. Accessed: 2026-06-04.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[7] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish pre-trained bert model and evaluation data, arXiv preprint arXiv:2308.02976 (2023).

[8] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), 2019, pp. 4171–4186.

[9] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, D. Zhou, Self-consistency improves chain of thought reasoning in language models, arXiv preprint arXiv:2203.11171 (2022).

[10] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, É. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised Cross-lingual Representation Learning at Scale, in: Proceedings of ACL 2020, 2020, pp. 8440–8451.