Text Is Not All You Need: Multimodal Prompting Helps LLMs Understand Humor - a Re-Do

Dimitar Dimov
Faculty of Computer Science and Engineering
dimitar.dimov@students.finki.ukim.mk

Abstract

While Large Language Models (LLMs) perform well on many text-based language tasks, understanding humor remains difficult. Humor is often multimodal: it depends not only on the words themselves, but also on how they are pronounced and spoken, including intonation. In this study, I investigate a simple multimodal prompting approach for humor understanding and explanation. I provide an LLM with both the written text of a joke and its spoken version, generated using an off-the-shelf text-to-speech (TTS) system. My results show that incorporating multimodal information leads to better humor explanations than text-only prompts across all evaluated datasets.

1 Introduction

Recent advances in Large Language Models (LLMs) have enabled strong performance across a wide range of natural language understanding tasks. However, humor understanding remains a challenging problem, particularly when linguistic ambiguity and phonetic cues play a central role. Prior work (e.g., Chum 2025 Study) has shown that providing LLMs with both textual and spoken representations of jokes can improve humor explanations, highlighting the importance of multimodal information.

In this work, I revisit this multimodal approach to humor understanding with a focus on reproducibility and accessibility. Instead of relying on proprietary or large-scale hosted models, I conduct all experiments using freely available, locally deployable LLMs. This setting allows us to examine whether previously reported gains from multimodal prompting persist under realistic resource constraints and with models commonly available to the research community.

I systematically compare text-only and text-plus-audio prompting across multiple humor datasets, with particular attention to pun-based jokes where phonetic ambiguity is critical. My results demonstrate that multimodal cues remain beneficial even for smaller, open-source models, supporting the robustness of the approach and underscoring the importance of speech information in computational humor understanding.

2 Methods

This study investigates whether access to audio information improves humor understanding and explanation in locally deployed, open-source language models. Due to the limited capacity of these models compared to large proprietary systems, a simplified multimodal prompting framework is adopted. Rather than few-shot or chain-of-thought prompting, all experiments use a zero-shot setup with controlled output formatting.

The overall pipeline consists of four stages: (1) explanation generation from text-only input, (2) explanation generation from combined text and audio input, (3) post-processing and normalization of model outputs into a strict JSON format, and (4) comparative evaluation using a separate judging model.

2.1 Audio generation

To expose pronunciation-based cues relevant to humor understanding, spoken versions of all jokes are generated using an off-the-shelf text-to-speech (TTS) system. Specifically, I use **Piper TTS**, a lightweight, open-source text-to-speech engine suitable for local deployment, to synthesize audio from the original joke text. No additional ground-truth information such as emphasis, timing, or prosody labels is provided. The generated audio is used solely as an auxiliary input intended to expose pronunciation-based ambiguity. All explanation prompts follow a zero-shot configuration. Each prompt includes a concise definition of puns and non-puns, along with instructions to determine whether the input text constitutes a pun and to explain the linguistic mechanism when applicable. Few-shot examples and explicit chain-of-thought prompting are omitted, as preliminary experiments showed that weaker local models struggled to maintain stable and consistent reasoning under such conditions.

Two explanation conditions are evaluated:

Text-only: The model receives only the written joke.

Text-plus-audio: The model receives both the written joke and its synthesized spoken form in a single prompt.

Unlike prior work that explored multimodal aggregation strategies, this study directly compares explanations generated under these two conditions. The prompt explicitly instructs the model not to reference the presence or absence of audio in its explanation, ensuring compatibility with text-only ground-truth annotations.

No additional ground-truth information such as emphasis, timing, or prosody labels is provided. The generated audio is used solely as an auxiliary input intended to expose pronunciation-based ambiguity.

2.2 Prompting

All explanation prompts follow a zero-shot configuration. Each prompt includes a concise definition of puns and non-

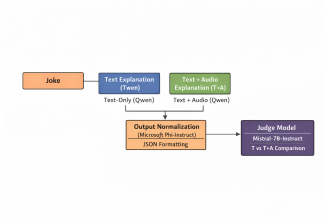

Figure 1: Strategy overview

puns, along with instructions to determine whether the input text constitutes a pun and to explain the linguistic mechanism when applicable. Few-shot examples and explicit chain-of-thought prompting are omitted, as preliminary experiments showed that weaker local models struggled to maintain stable and consistent reasoning under such conditions.

Two explanation conditions are evaluated:

Text-only: The model receives only the written joke.

Text-plus-audio: The model receives both the written joke and its synthesized spoken form in a single prompt.

Unlike prior work that explored multimodal aggregation strategies, this study directly compares explanations generated under these two conditions. The prompt explicitly instructs the model not to reference the presence or absence of audio in its explanation, ensuring compatibility with text-only ground-truth annotations.

2.3 Output normalization and evaluation

All generated explanations are converted into a strict JSON format to enforce consistency and enable automated comparison. Outputs that fail to conform to this format are marked as invalid.

To assess explanation quality, a separate judging model is employed. The judge is prompted to compare paired explanations produced from the same joke under the text-only and text-plus-audio conditions and to determine which explanation more accurately identifies whether the input is a pun and better explains its linguistic basis. This comparison-based evaluation isolates the effect of audio input while controlling for model architecture and prompting strategy.

3 Experimental Setup

3.1 Datasets

I evaluate the proposed approach on three humor datasets that vary in size, annotation quality, and availability of human-written explanations.

SemEval 2017 Task 7. This dataset contains 810 homographic puns, 647 heterographic puns, and 1,077 non-puns. The puns are clearly separated by type. This annotation makes SemEval particularly suitable for evaluating explanation quality and the underlying mechanisms of puns.

Context-Situated Puns. This dataset consists of 821 homographic and 1,739 heterographic puns with human annotations specifying the pun word and its interpretations sun2022context. Unlike SemEval, it does not provide human-written explanations, allowing evaluation to focus on the model's ability to infer and explain linguistic ambiguity without reference explanations.

ExplainTheJoke. This dataset contains 377 jokes collected from the ExplainTheJoke.com website. Each entry includes a human-written explanation in paragraph form, but no structured annotations. These explanations exhibit high variability in length, style, and accuracy, making the dataset suitable for evaluating explanation robustness in less controlled, real-world settings.

3.2 Models

All experiments are conducted using freely available, locally deployable language models.

Explanation generation. Both text-only and text-plus-audio explanations are generated using **Qwen2-Audio**, which supports audio-conditioned generation while remaining deployable in a local environment. The same model and prompt structure are used for both conditions to ensure a controlled comparison, with the only difference being the presence or absence of audio input.

Output normalization. Generated explanations are normalized into a strict JSON format using **Microsoft Phi-Instruct**. This normalization step ensures consistent structure across outputs and enables automated downstream evaluation. Outputs that fail to conform to the required format are marked as invalid and excluded from further analysis.

Evaluation (judge model). Explanation quality is assessed using **Mistral-7B-Instruct-v0.3**, employed as an independent judging model. The judge compares paired explanations generated from the same joke under text-only and text-plus-audio conditions and determines which explanation more accurately identifies whether the input is a pun and better explains its linguistic mechanism. Using a separate model for evaluation avoids biases that may arise from judging a model's own outputs and follows recent work supporting LLM-based evaluation of free-form text generation.

4 Results

4.1 SemEval

For the SemEval 2017 Task 7 dataset, evaluation is conducted using a comparison-based judging framework. A balanced subset of 250 homographic and 250 heterographic puns is sampled from the dataset. For each joke, two explanations are generated: one using text-only prompting and one using text-plus-audio prompting.

An independent judging model is prompted with the original joke text and the two explanations and tasked with determining which explanation more accurately identifies whether the text is a pun and better explains the underlying linguistic mechanism. The judging instructions explicitly discourage positional bias and verbosity bias, and emphasize correctness, clarity, and accurate identification of wordplay. When both explanations are of comparable quality, the judge may assign a tie.

The judge outputs its decision in a strict JSON format, indicating whether Explanation 1 is superior, Explanation 2 is

SemEval Split	Text wins	Audio wins	Tie	Invalid
heterographic	78 (31.2%)	109 (43.6%)	1 (0.4%)	62 (24.8%)
homographic	92 (36.8%)	96 (38.4%)	6 (2.4%)	56 (22.4%)

Table 1: SemEval results under pairwise judging. "Text wins" indicates the judge preferred the text-only explanation (Explanation 1), while "Audio wins" indicates preference for the text-plus-audio explanation (Explanation 2). "Invalid" denotes cases where outputs failed to conform to the required JSON format.

superior, or whether both explanations are of similar quality, along with a brief justification. No audio input is provided to the judge; evaluation is based solely on the textual explanations and the original joke text.

4.2 Context-Situated Puns

For the Context-Situated Puns dataset, evaluation is conducted using the same comparison-based judging framework. A subset of 500 examples is sampled from the dataset, covering both homographic and heterographic puns. For each example, two explanations are generated: one using text-only prompting and one using text-plus-audio prompting.

An independent judging model is provided with the original joke text and the two generated explanations and tasked with determining which explanation better identifies the presence of a pun and explains the underlying linguistic ambiguity. The judge is explicitly instructed to avoid positional bias and verbosity bias, and to prioritize correctness, clarity, and accurate identification of wordplay. When both explanations are of comparable quality, the judge may assign a tie.

As in all experiments, the judge does not receive audio input; evaluation is based solely on the textual explanations and the original joke text.

Context-Situated Puns (N=299)		
Pairwise preference (from judge scores)		
Tie	232	77.6%
Text > Audio	53	17.7%
Audio > Text	14	4.7%
Score statistics (1–5 scale)		
Text mean / median	3.696 / 3.0	
Audio mean / median	3.428 / 3.0	

Table 2: Results on Context-Situated Puns. Preferences are derived by comparing the judge-assigned 1–5 scores for text-only vs. text-plus-audio explanations; equal scores are counted as ties.

4.3 ExplainTheJoke

To evaluate explanation quality beyond puns, the ExplainTheJoke dataset is used in its entirety. This dataset contains jokes spanning a wide range of humor types and does not provide structured annotations. For each joke, explanations generated under the text-only and text-plus-audio conditions are evaluated independently rather than through direct pairwise comparison.

An independent judging model is prompted with the joke text and a single generated explanation and asked to assess how well the explanation accounts for why the joke is humorous. The judge assigns a score on a five-point scale based on correctness, clarity, and accurate identification of the humor mechanism, while penalizing hallucinated or irrelevant reasoning. As in all experiments, the judge does not receive audio input. The results show that explanations generated with access to audio receive slightly lower quality scores overall, indicating that audio information can sometimes confuse the model for jokes that do not rely on explicit phonetic ambiguity.

Preferred Explanation	Count	Percentage
Text-only	63	16.7%
Text + Audio	39	10.3%
Tie	275	72.9%

Table 3: Pairwise evaluation results on the ExplainTheJoke dataset.

5 Analysis and Conclusion

Across datasets, the effect of adding audio is mixed and appears to depend on both humor type and the stability of local generation. On SemEval, text-plus-audio explanations are preferred more often overall for heterographic puns, consistent with the importance of pronunciation-based ambiguity, while homographic puns show a more balanced outcome, reflecting the limited utility of audio when ambiguity is primarily orthographic. In contrast, on Context-Situated Puns, most comparisons result in ties (77.6%), and in non-tied cases text-only explanations are preferred substantially more often than audio-conditioned explanations (17.7% vs. 4.7%); the judge's mean score is also higher for text-only outputs (3.696 vs. 3.428). For ExplainTheJoke, the majority of cases are again judged as ties, and text-only explanations are preferred more often than audio-conditioned explanations in non-tied cases, indicating that audio provides limited or inconsistent benefits for broader, non-pun humor. Taken together, these findings suggest that multimodal prompting with locally deployable models can help in pronunciation-dependent settings (e.g., heterographic puns), but is not universally beneficial and may introduce noise or reduce explanation quality when phonetic cues are not central or when synthesized speech does not add discriminative information.

References

6 Related Work

- Text Is Not All You Need: Multimodal Prompting Helps LLMs Understand Humor (CHUM 2025)
- Multimodal Sarcasm Recognition by Fusing Textual, Visual and Acoustic Content via Multi-Headed Attention
- Lexical and syntactic ambiguity as a source of humor: The case of newspaper headlines

- SemEval 2017 Task 7: Pun Detection and Interpretation (Miller et al., 2017)
- Qwen documentation
- Mistral documentation
- Microsoft PhiCookBook
- GNN Pun Detection and Location repository
- Piper TTS documentation