CS8803: Project Report

Explaining Puns: Enhancing Computational Humor Understanding with Large Language Models

Yingchen Ma, Nemath Ahmed, Rajvi Parekh

1 Introduction

Despite significant advances in Large Language Models (LLMs), such as GPT-3 and GPT-4, computational humor remains a challenging area [6]. These models often struggle to accurately assess and generate humor, misinterpreting context and cultural nuances, and recycling jokes [5]. Existing literature has identified limitations in LLMs' humor assessment and generation capabilities, including difficulties in recognizing and generating humor within specific themes, interpreting humor across diverse cultural contexts, and avoiding maladaptive humor [2]. As humor is commonplace in human conversation and society, the ability of LLMs to understand and interpret humor is an area that is valuable to explore.

Our project aims to address these gaps by introducing an evaluation framework for humor in LLMs, focusing on a specific type of humor: puns. By developing this framework, we aim to work towards a set of metrics and benchmarks for comprehensively assessing humor in LLMs, benchmark LLM performance against human humor understanding, and provide actionable insights for refining LLMs' humor capabilities.

To achieve this, alongside the evaluation framework, we are employing two approaches: (1) fine-tuning, and (2) few-shot prompting, to enhance the ability of LLMs to generate explanations for puns. We have trained our model and fine-tuned it using a supervised fine-tuning approach. We have developed an evaluation framework that includes generating explanations for jokes and assessing the model's understanding of humor using quantitative and qualitative methods. We developed a quantitative method to evaluate explanation quality by calculating cosine similarity between the model's output and (human created) ground truth explanations, alongside a parallel qualitative method consisting of four evaluation metrics judged directly by human annotators. Our work aims to bridge the gap between LLMs' current humor capabilities and human humor standards, paving the way for more effective and nuanced computational humor models.

2 Prior Work

Recent studies have delved into the challenges surrounding humor assessment and generation in Large Language Models (LLMs). Sun et al. [6] scrutinized the humor capabilities of ChatGPT in humor comprehension and generation; their results shed light on its limitations. They found that while ChatGPT can communicate on a human-like level, its ability to generate novel jokes is constrained, often recycling a limited set of jokes and struggling with contextual nuances. Our work extends beyond theirs by introducing a more comprehensive evaluation framework for humor understanding that can be generalized across diverse contexts and types.

Similarly, Goes et al. [5] investigated the ability of GPT-4 to evaluate jokes compared to human judgments. Their work focused on creating system descriptions in GPT-4 to mimic human judges' evaluation of humor. While their approach provided insights into LLMs' ability to evaluate jokes, the prior work primarily focused on subjective, individual scores of "funniness" rather than more grounded and objective assessments; the latter is important because humor can be highly subjective. Our project builds upon their system description methodology by conducting fine-tuning and few-shot prompting approaches. Our evaluation framework is also more comprehensive as the prior work is more subjective; they use the average funniness score rated by humans and the score generated by GPT-4 for each system description. In contrast, our evaluation framework is more objective with the use of cosine similarity and incorporates multiple prompting techniques.

Furthermore, Bogireddy et al. [2] addressed the challenges of humor detection in the specific context of humor related to the COVID-19 pandemic. Their dataset, comprising various humor sources such as Reddit posts, news headlines, and tweets (from Twitter/X), illustrated the training and evaluation of humor detection models. While their work highlighted improvements in humor detection with fine-tuned models, it primarily focused on detecting humor, a simpler task than explaining humor. Our project complements their efforts by focusing on the assessment of humor.

In summary, while existing studies have contributed valuable insights into the challenges of computational humor, our project aims to address their limitations by introducing a more comprehensive evaluation framework and exploring multiple approaches to improve the humor explanation ability of LLMs. By doing so, we seek to advance the field of computational humor and contribute to the development of more sophisticated LLMs with enhanced humor capabilities.

3 Dataset

For our project, we utilized the ExPUNations dataset [7], a comprehensive dataset consisting of puns with extensive, finely-grained annotations. This dataset was created for two primary purposes: (1) aiding the understanding of puns through explanation generation, and (2) facilitating keyword-conditioned pun generation. Its design caters to these tasks, providing detailed annotations and facilitating a deeper understanding of puns.

3.1 Dataset Overview

Figure 1 portrays an overview of our dataset, which comprises pun jokes and their corresponding natural language explanations. Each sample in the dataset contains a joke that is potentially a pun, alongside exactly five natural language explanations created by human annotators that provide insight into why the joke is humorous. Figure 2 shows a few examples of (full) puns and explanations in the dataset.

The original dataset, as downloaded, contained 1,899 unique jokes with 9,495 explanations (exactly 5 explanations per joke). After filtering for annotations that mark the joke as a joke, the dataset was reduced to 1,739 unique jokes and 6,279 explanations. Further filtering for jokes where all five annotators agree it is a joke resulted in 575 unique jokes and 2,875 explanations.

Additionally, Figure 3 shows the distribution of joke lengths (defined as the number of words) in the dataset. The majority of pun jokes are of short to medium length, with very few extending beyond 25 words.

	ID	text	Natural language explanation 1	Natural language explanation 2	Natural language explanation 3	Natural language explanation 4	Natural language explanation 5	model_inference_input_text
0	het_1035	When bottled water is cheap it's called a liqu...	A liquidation sale is when the price is greatl...	Liquid is found in "liquidation". In business,...	Water is a liquid. Liquidation is the process ...	This is a play on words. The term "liquidation...	liquid is in the word liquidation and water is...	<s>[INST] Provide an explanation on why this p...
1	het_1213	If intervening was an olympic sport , he'd win...	To 'meddle' is to mess around, or intervene, i...	This is a pun on how 'meddle' sounds like 'medal'	The joke is a pun. 'Meddle' means to interfer...	This is a pun on 'meddle' because it both soun...	meddle sounds like medal	<s>[INST] Provide an explanation on why this p...
2	het_1221	I'm always exhausted by Friday , said Tom weak...	Weakly means exhausted or without strength, an...	"Weakly" is a homophone of "weekly". Weekly me...	The joke is implying that the airplane flew	weakly sounds like weekly if he's visiting the...	This is a play on words. The word "weakly" mea...	<s>[INST] Provide an explanation on why this p...
3	het_1249	The math teacher was a good dancer - he had al...	An algorithm is a function in mathematics, whi...	In mathematics, an algorithm is a sequence of ...	Algorithm, alluding to math, is being used to ...	algorithm sounds like rhythm	This is a play on words. The word "rhythm" mea...	<s>[INST] Provide an explanation on why this p...

Figure 1: Overview of Dataset

ID	text	Natural language explanation 1	Natural language explanation 2	Natural language explanation 3	Natural language explanation 4	Natural language explanation 5
het_100	I phoned the zoo but the lion was busy .	lion sounds like line	This is a pun on the phrase 'the line was busy' which means that the telephone line was currently engaged in another call. However, 'lion' is used in it's place because it sounds similar and lions are an animal found in zoos	The joke is a pun. Lions may be typically seen at a zoo. 'Lion' sounds like 'line' and to say 'the line was busy' means that the phone call would not go through due to too many people calling it already.	Lion sounds like line, as in phone line, and a lion might be commonly found at a zoo.	This is a somewhat amusing mental image on top of being a pun. 'lion' is a pun for [phone] line on top of being a zoo animal.
het_1006	I heard Einstein got along well with his parents . . . relatively speaking .	This is a pun on Einstein's Theory of Relativity.	Einstein invented the theory of relatively and got on relatively well with his parental relatives.	This is a pun on 'relatively' which is initially referring to Einsteins relation to his parents as they are related, but its also being used since Einstein came up with the theory on relativity	The joke is a pun. 'Relatively speaking' means talking about one thing in relation to another. Einstein came up with the theory of relativity.	relatively can refer to his theory of relativity
het_1007	The salt said'hi'to the pepper . It was seasonings greetings .	This pun is replacing season with seasoning, as in seasons greetings which is a phrase often used to express salutations during the holiday season - all in the context of salt and pepper greeting eachother.	This is a play on the phrase "seasons greetings." salt and pepper are both seasonings	The text "seasonings greetings" sounds like "seasons greetings" which is a common phrase used during the holiday season on cards. Salt and pepper are types of seasonings. Salt and pepper are getting personifed and greeting each other.	This is a play on words. The phrase "season's greetings" is used as an expression of goodwill during winter holidays but "seasonings greetings" refers to the fact that seasonings, or spices added to foods, are greeting each other.	salt and pepper are considered seasonings

Figure 2: Example Puns and Explanations

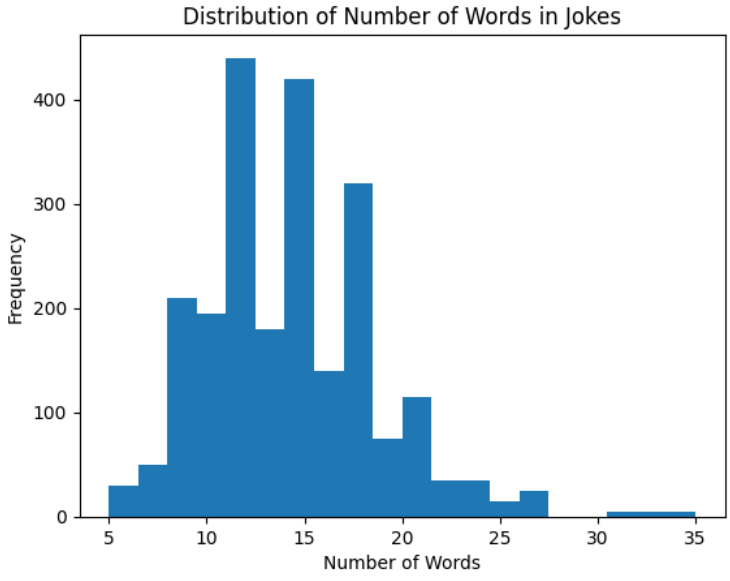

Figure 3: Distribution of Number of Words in Jokes

3.2 Data Preprocessing

We filter and preprocess the original dataset for the purposes of our project. The original dataset consisted of 1,899 unique jokes and 9,495 explanations. However, the dataset consist of a column that represents a human annotator's opinion of whether or not the pun constitutes a valid joke in the first place. As we want puns that are generally agreed to be valid jokes for our experiments, we only keep jokes where all five annotators marked it as a joke - resulting in 575 unique jokes and 2,875 explanations. From here, we only use the "text" and the five "Natural language explanation" columns; there exist other human-annotated columns in the original dataset but they are not relevant for our project.

4 Project Approach

In this section, we discuss the methodology that we employed for this project. In summary, we take two approaches to try to enhance the pun explainability performance of large language models: (1) fine-tuning on our dataset of puns annotated with explanations of each of the puns, and (2) doing few-shot prompting using a set of selected pun-explanation examples.

For our experiments, we perform a 80-20 train-test split of the ExPUNations dataset; we use the training set exclusively for fine-tuning, and the testing set exclusively for evaluation for all settings. The training set consists of 460 unique jokes with 2,300 explanations, while the testing set comprises 115 unique jokes with 575 explanations.

4.1 Fine-Tuning Approach

For the fine-tuning approach, we present pun-explanation pairs from the ExPUNations training set, and fine-tune for 1 epoch. Additionally, we experiment with two different prompts for the fine-tuning to understand the impact of using two different prompts:

1. Prompt 1:
```
Provide an explanation on why this
pun is funny.
```

2. Prompt 2:
```
You are a bot that specializes in
analysis of humor. You specialize
in exploring the wordplay, setup of
the joke, the punchline, the comedic
techniques used, puns, and plays on
words. Provide an explanation on why
this joke is funny.
```

Prompt 1 is a short prompt that presents the pun explanation task to the model in very basic and simple terms. Prompt 2 is a longer prompt that also presents to the model a system description (following the approach in [5]), in an effort to help the model better grasp the task.

4.2 Few-Shot Prompting Approach

For the few-shot prompting approach, we present examples alongside the task description. We use two few-shot prompting settings: 3-shot and 7-shot, to investigate if the number of examples presented affects the performance. The text format for an n-shot input is:

```
Provide an explanation on why this pun is funny.
When answering, follow these examples:
Pun: <pun 1>
Explanation: <explanation 1>

Pun: <pun 2>
Explanation: <explanation 2>

...

Pun: <pun n>
Explanation: <explanation n>
```

To select examples from the ExPUNations dataset, one member of our project shortlisted a set of 10 examples that were determined to provide the LLM a good understanding of the ideal output explanation that we'd expect. We also based it off the criterion in Section 5.2 and shortlisted ones which are clear, complete, relevant and insightful based on general understanding. From here, the other two members of our project applied the same criteria to the shortlisted examples, resulting in a total of 7 examples that all three members agreed on. These 7 examples were selected to be included in the model input; all 7 for the 7-shot setting, and a subset of 3 for the 3-shot setting. Table 2 shows each pun-explanation example.

Table 1: Full list of training parameters

Parameter	Value
Train/test split	80/20
Optimizer	Paged AdamW (32-bit)
Learning rate	2e-4
Weight decay	0.001
Maximum gradient normal	0.3
# of epochs	1
Training batch size	4
LoRA attention dimension (r)	64
LoRA scaling parameter (α)	16
LoRA dropout probability	0.1

4.3 Experimental Settings

For our model, we selected Llama-2-7b, which has recently demonstrated state-of-the-art performance in dialogue use cases [8]. We ran all experiments using Google Colab, on a T4 GPU available under a Colab Pro subscription. In order to enable training and inference of the model on the T4 GPU (15 GB GPU memory) on Colab, we make use of QLoRA [3], which quantizes the 16-bit model into 4-bit during training time with minimal performance degradation. A full list of training parameters (identical for all experiments) can be found in Table 1.

5 Evaluation

In this section, we will discuss the evaluation methods of our experiments. We evaluate our methods in two ways: both quantitatively and qualitatively. For the former, we employ methods to automatically calculate the semantic similarity of model-produced explanations to human-produced explanations. Though they can be done on a larger scale, automatic metrics may not capture all of the nuances in the produced results. As a result, we also perform qualitative evaluation in order to collect direct human feedback about the quality of the produced explanations. Both evaluations are done exclusively on the testing set.

5.1 Quantitative Evaluation

For each pun, we compute the average cosine similarity between the produced model explanation and each of the five human-produced explanations. To accomplish this, we create sentence embeddings generated by a pre-trained BERT [4] model fine-tuned for natural language understanding tasks. The process involves encoding each sentence into a high-dimensional vector representation using the BERT tokenizer and model. We calculate the cosine similarity between the produced model explanation and each of the five human-produced explanations. The cosine similarity metric measures the cosine of the angle between two vectors and ranges from -1 to 1, where 1 indicates perfect similarity and -1 indicates perfect dissimilarity. By averaging the cosine similarities across all five human-produced explanations for each pun, we obtain a comprehensive measure of how closely the model-generated explanation aligns with human explanations.

$$\text{Cosine Similarity}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|} \tag{1}$$

In this equation:

- $\mathbf{u}$ and $\mathbf{v}$ represent the sentence embeddings for the model-produced explanation and the human-produced explanation, respectively.
- $\cdot$ denotes the dot product operation.
- $\|\mathbf{u}\|$ and $\|\mathbf{v}\|$ denote the Euclidean norms of vectors $\mathbf{u}$ and $\mathbf{v}$, respectively.

5.2 Qualitative Evaluation

Our method for qualitative analysis was designed to ensure a thorough evaluation of the generated explanations for puns. We considered human evaluations of key factors that contribute to the overall quality of an explanation, namely: clarity, completeness, relevance, and insightfulness. Human evaluation provides a comprehensive contextual benchmark beyond the task, domain, and semantic capabilities oriented current technical approaches [1].

The four human evaluation metrics we use are:

1. Clarity: We evaluate whether the explanation is clear and easy to follow. Clarity ensures that the generated explanations are easily understandable and coherent. A clear explanation is essential for ensuring that the humor contained within the pun is effectively communicated.
2. Completeness: We assess whether the explanation covers all aspects of the pun and how it works. By evaluating completeness, we ensure that the generated explanations are comprehensive, leave no aspect of the pun unexplained, and exhibit a thorough understanding of the humor.
3. Relevance: We determine if the explanation is relevant to what the pun is discussing. Relevance ensures that the generated explanations are contextually appropriate and directly related to the pun, and do not contain unnecessary information outside of that scope.
4. Insightfulness: We evaluate whether the explanation demonstrates a deep understanding of the pun and its meaning or context. Insightfulness ensures that the generated explanations provide valuable insights into the underlying humor of the pun, and encourages the generated explanations to go beyond surface-level analysis and provide a deeper understanding of the humor.

By employing this qualitative analysis method, we ensure a comprehensive and rigorous evaluation of the quality of the generated explanations. This method allows us to assess the effectiveness of our model in explaining pun jokes and provides valuable insights into its performance. Each measure is rated on a 5-point Likert scale (1 = least, 5 = most).

5.2.1 Annotation Process

For the annotation process, each of the 115 pun explanations generated by the model before fine-tuning, along with the 115 explanations generated after fine-tuning, were independently evaluated by two annotators. Annotators rated the quality of each explanation based on the four metrics. This approach ensured a thorough, robust, and consistent evaluation of the model's output. Each of the 230 (115 + 115) explanations produced received a total of two scores for each metric. Each of the scores for each explanation-metric were averaged together to produce the final four scores for each pun before and after finetuning.

6 Results and Discussion

6.1 Quantitative

Figure 4 shows the results of the quantitative experiments, including both the fine-tuning experiments (with Prompts 1 and 2), and the few-shot experiments (with Prompt 1).

The fine-tuning experiments show that finetuning was effective at making the model produce output explanations that more closely resemble those produced by humans, as seen by the increases in cosine similarities for both prompts. Prompt 1 achieved a median average cosine similarity of 0.7782 before finetuning and 0.8255 (the highest of all of our experiments) after finetuning. Meanwhile, Prompt 2 achieved a median cosine similarity of 0.7982 before finetuning and 0.8120 afterward.

Prompt 2, which contains a system description, seems to have been more effective at producing more human-realistic explanations without finetuning than Prompt 1, which did not contain a system description. Meanwhile, after the finetuning, Prompt 1 performed better; a possible reason is that the simpler prompt more closely resembles the instructions given to the annotators of the original dataset.

Regarding the few-shot experiments using Prompt 1, both the 3-shot and 7-shot settings achieved average cosine similarities within the range of the fine-tuning experiments: 0.8104 and 0.7965, respectively. The 3-shot setting performed better than the 7-shot setting; this is a counter-intuitive finding, as one may expect more examples to enhance the model's understanding.

6.2 Qualitative

Figure 5 shows the results of the qualitative experiments. We use the finetuning approach with Prompt 1 for all the qualitative experiments, as it produced the highest median average cosine similarity (0.8255) for the qualitative experiments.

For all four human evaluation metrics, the median scores for each metric increased from before finetuning to after finetuning. The increase was the greatest for clarity (3.0 - 4.5) and completeness (3.5 - 4.5). The majority of the 1st and 3rd quantile scores for each metric increased as well. Overall, these results demonstrate clear evidence that model finetuning for humor explanation using human-created annotations was able to produce explanations that were easier to understand and more effective at communicating the humor in the pun. This also reinforces the high quality of the human-created explanations in the original dataset.

7 Conclusion

In conclusion, our study presents novel strategies for enhancing computational humor understanding, specifically in the domain of pun jokes, using large language models. Through fine-tuning and few-shot prompting techniques, we observed improvements in the model's ability to generate explanations that closely resemble human-produced ones and are more effective in communicating the humor of the pun.

While our results are promising, there do exist some limitations and related avenues for future work. First, we only experiment on one model; future work can explore other open-source models alongside those that are not open source. In addition, the four evaluation metrics were conceptualized by the project members for this work; future work can explore more, possibly shaped

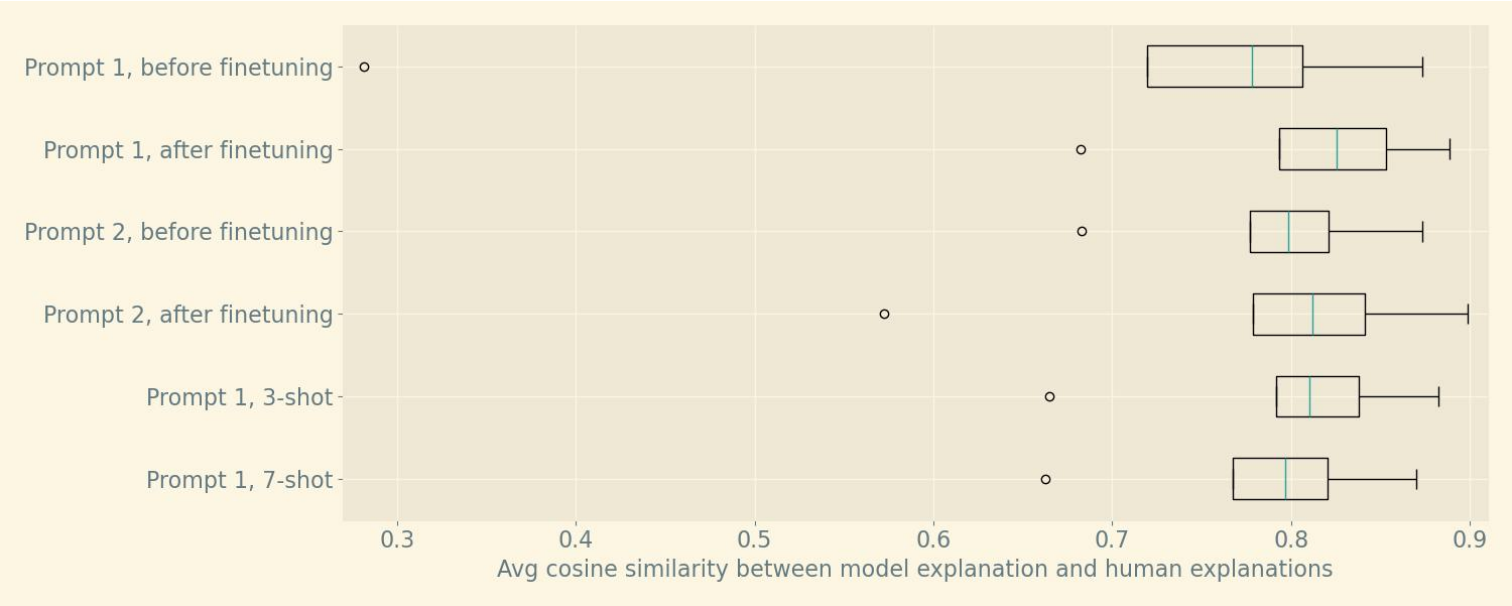

Figure 4: Boxplots showing the results of the quantitative experiments. Each boxplot shows the minimum, median (blue), maximum, and quantiles.

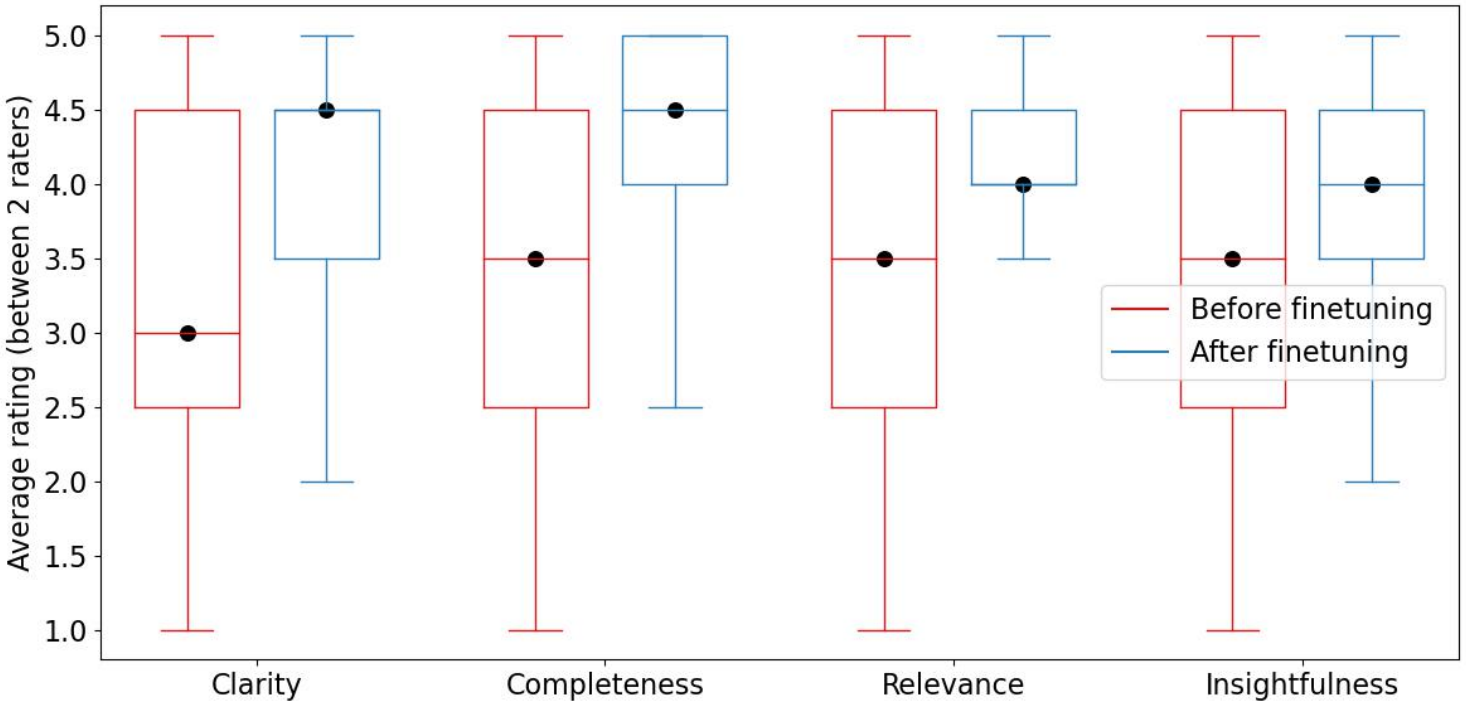

Figure 5: Boxplots showing the results of the qualitative experiments, before and after finetuning with Prompt 1, for all four evaluation metrics. Each boxplot shows the minimum, median (black dot), maximum, and quantiles.

in collaboration with experts in the domains of humor and computational humor. Also, larger-scale human subject studies can be employed to evaluate the effectiveness of AI-generated joke explanations. Finally, jokes beyond puns can be explored, particularly types with more complex structures and more diverse contexts.

Overall, our work contributes to advancing the field of computational humor and lays the groundwork for more nuanced and effective humor generation models.

References

[1] Raghav Awasthi, Shreya Mishra, Dwarikanath Mahapatra, Ashish Khanna, Kamal Maheshwari, Jacek Cywinski, Frank Papay, and Piyush Mathur. Humanely: Human evaluation of llm yield, using a novel web-based evaluation tool, 12 2023.

[2] Neha Reddy Bogireddy, Smriti Suresh, and Sunny Rai. I'm out of breath from laughing! i think? a dataset of covid-19 humor and its toxic variants. In *Companion Proceedings of the ACM Web Conference 2023*, WWW '23 Companion,

Table 2: The list of 7 examples selected for the few-shot prompting approach. Examples 1-3 are used for the 3-shot approach, while all 7 are used for the 7-shot approach.

Example No.	Pun	Explanation
1	I phoned the zoo but the lion was busy.	This is a pun on the phrase 'the line was busy' which means that the telephone line was currently engaged in another call. However, 'lion' is used in its place because it sounds similar and lions are an animal found in zoos.
2	I keep reading 'The Lord of the Rings' over and over. I guess it's just force of hobbit.	Force of habit means something that has the tendency for something to be done frequently. Lord of the Rings is a series of fantasy novels written by J.R.R. Tolkien. 'The Hobbit' is a fantasy novel that proceeded the Lord of the Rings series which was also written by J.R.R. Tolkien. The joke is centered around the word 'hobbit' since it sounds like 'habit' and its use in the common phrase 'force of habit', turned into 'force of hobbit'.
3	The fisherman kept bragging about the big fish he caught, but he would not be very pacific about where he caught it.	Pacific is the name of an ocean. An ocean is where you can catch fish. 'Pacific' sounds close to 'specific'. Since the joke is discussing a fishman catching fish, the word 'specific' was replaced with 'pacific' which relates to an ocean where fish can be found. The joke is playing on the word 'pacific' because it sounds close to 'specific'.
4	Thieves have muscles of steal.	Steal and 'steel' are homophones. Steel is a type of metal. To say someone has 'muscles of steel' is to say that they are very strong. Thieves are people that steal when they want something. To steal is to take something of another's without permission. The joke is playing on the words 'steal' and 'steel'.
5	I met a man who loves eating couches. I think he has a suite tooth.	A suite is a set of rooms that contain furniture. A couch is a piece of furniture that multiple people can sit on. Suite and sweet are homophones. Sweet is a descriptive word used to describe the taste of something, typically if that something is sugary. If someone has a 'sweet tooth' it means they really like and crave eating sweet things. The joke is centered around the word suite and its homophone sweet.
6	When asked by her co-workers whether they should bring a gift to her birthday party, Mary replied, 'You should know that all I'm interested in is your presence'.	This is an amusing joke that seems to imply opposite meanings - making it a double entendre. If Mary only cares about their presence, it means she just wants her guests to show up, but the word itself sounds a lot like presents, implying the exact opposite of what was written.
7	A politician who had been an astronomer was always saying 'no comet'.	This is a pun on the phrase 'no comment' which is usually said by politicians when they wish to not comment on a cetain topic, but 'comet' is used in its place because it sounds similar and because comets are a celestial object consisting of a nucleus of ice and dust and, when near the sun, a tail of gas and dust particles pointing away from the sun, which is something that astronomers study.

page 1004–1013, New York, NY, USA, 2023. Association for Computing Machinery.

[3] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[5] Luis Fabricio Góes, Piotr Sawicki, Marek Grzes, Dan Brown, and Marco Volpe. Is GPT-4 Good Enough to Evaluate Jokes? 6 2023.

[6] Sophie Jentzsch and Kristian Kersting. Chatgpt is fun, but it is not funny! humor is still challenging large language models, 2023.

[7] Jiao Sun, Anjali Narayan-Chen, Shereen Oraby, Alessandra Cervone, Tagyoung Chung, Jing Huang, Yang Liu, and Nanyun Peng. ExPUNations: Augmenting puns with keywords and explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

[8] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.