JokeEval: Are the Jokes Funny? Review of Computational Evaluation Techniques to improve Joke Generation

Sulbha Jain
suljain@amazon.com
Amazon
Seattle, WA, USA

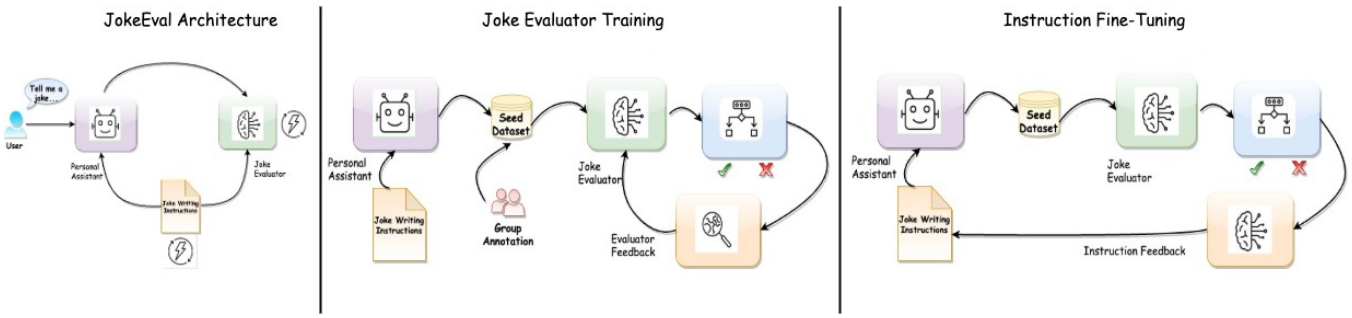


Figure 1: JokeEval: Architecture for Training and Scoring pipeline

Abstract

Humor is a complex yet essential aspect of human communication. It can be defined as a communicative expression establishing surprising, incongruent relationships or meanings to amuse. This paper presents empirical evidence demonstrating the successful application of computational methods to humor recognition in AI generated textual data, specifically jokes. Through experiments on synthetic and open-source datasets, we show that automatic classification techniques can effectively differentiate between "Funny" and "Not Funny" jokes. Our results reveal that hybrid Convolutional Neural Networks with Recurrence, trained on high-dimensional vector embeddings of synthetic jokes, achieve a statistically significant F1-Score of 71.2% on the ColBERT dataset. These findings underscore the potential of machine learning approaches in capturing the nuanced nature of humor, paving the way for more sophisticated computational understanding of this fundamental aspect of human interaction and providing a feedback loop for funnier joke generation.

CCS Concepts

• **Humor Evaluation**; • **Scalable**; • **LLM Evaluation**; • **AI Content Generation**; • **AI Evaluation**;

KDD '25, Toronto, ON, Canada

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Sulbha Jain. 2025. JokeEval: Are the Jokes Funny? Review of Computational Evaluation Techniques to improve Joke Generation. In *Proceedings of KDD (KDD '25).* ACM, Toronto, ON, Canada, 17 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Humor represents a fundamental aspect of human communication that emerges when unexpected elements are combined in ways that trigger cognitive shifts leading to laughter [8, 25]. The complexity of humor stems from its basis in incongruity, when information or situations are presented in inconsistent or disharmonious ways [12, 30]. Individual appreciation and expression of humor varies significantly based on sociocultural contexts, mood, and personal experiences, making humor recognition and analysis particularly challenging yet fascinating for computational study [32].

The field of computational humor has become increasingly relevant in AI research, with potential to transform AI systems into more creative and motivational tools [5]. As AI systems advance rapidly in their content generation capabilities, including joke creation, there is a pressing need for robust frameworks to evaluate and assess AI-generated humor [10, 14, 24]. While jokes can be defined as short humorous pieces culminating in a final punchline [17], the significant difference between spontaneous real-world humor and scripted scenarios presents a critical challenge in objectively measuring the comedic effectiveness of AI-generated jokes [11]. This discrepancy presents a critical challenge in objectively measuring the comedic effectiveness of AI-generated jokes.

This research specifically focuses on detecting humor opportunities in English language jokes delivered by virtual personal agents [14, 31]. The study's importance is highlighted by users' frequent requests for jokes from virtual assistants, where inadequate humor delivery can lead to user dissatisfaction and abandonment [34]. The proposed approach, which can be applied universally

across languages and personal assistants, aims to automate funnier joke generation through:

- Using supervised machine learning algorithms for joke pattern recognition.
- Experimenting with various Deep Neural Network variants, including a novel Hybrid Convolutional and Recurrent-based Joke Classifier Model.
- Testing and analyzing Large Language Models (LLMs) for humor detection and providing feedback for improvement.

The rest of the paper is structured as follows: Section 2 discusses related work on humor recognition with a focus on one-liner jokes and ML classification techniques, Section 3 proposes experiment design for computational humor recognition along with data preparation, Section 4 presents performance results, and Section 5 concludes with future work directions.

2 Related Work

The detection of computational humor presents greater challenges than humor generation, which typically relies on templates [37]. Early research attempts focused on analyzing structured texts like knock-knock jokes using n-grams, though these approaches were limited in scope [28, 29]. While recent approaches using BERT-based models and Convolutional Neural Networks (CNNs) show promise, they lack extensive experimental validation [4, 6, 26]. In cases where template-based detection is not feasible, researchers have explored machine learning classification methods to distinguish between humorous and non-humorous text by analyzing contrasting features [21, 23]. These studies primarily concentrated on detecting one-liner jokes, which are defined as short sentences with comic effects and interesting linguistic structures. Notable work by Mihalcea and Pulman [20] demonstrated successful one-liner detection using Na"ive Bayes and Support Vector Machine algorithms for text classification. Domain-specific research has been conducted in areas such as customer reviews, drawing from Berger's topology [19]. Recent studies have explored using Large Language Models (LLMs) for genre-specific humor detection [35], though with limited experimentation on open-source data. Research has also included computationally intensive LLM fine-tuning with custom joke generators [36]. A significant recent advancement in the field is the THInC framework, which helps bridge the gap between humor theory research and computational humor detection [6].

Previous evaluations of humor detection have been constrained by narrow domains and small sample sizes, lacking comprehensive analysis and confidence rubrics derived from broader understanding of jokes using LLMs. This limitation highlights the need for more robust and generalizable approaches to computational humor detection.

3 Experiment

In order to run the experiments as seen in Figure 1 we prepare the data as per Section 3.1. Experiment results are measured as per metrics listed in Section 3.2. We discuss the detailed model designs in Section 3.3.

3.1 Data Preparation

3.1.1 Train Dataset. Our data preparation process consisted of several key steps ensuring a diverse and balanced dataset for training our humor detection models:

Synthetic Dataset Generation. We curated our seed dataset (synthetic jokes) using `Claude Sonnet 3.5` [2] with the following parameters: $Temperature : 1.0, Top - p : 0.1, Top - k : 250$ for higher creativity and response variability. We instructed the LLM to generate short, funny jokes as per the prompt detailed in Appendix Section .5.

Human Annotation. The AI-generated jokes were evaluated for humor through a manual annotation process involving multiple human assessors. These evaluators were tasked with categorizing each joke as either "Funny" or "Not Funny" and were compensated at a rate of $50 per hour. Each joke was assessed by three different annotators, with a tie-breaking process in place to resolve any discrepancies in ratings and overcome personal biases. Our annotation system used a binary scoring method, where jokes considered funny were assigned a score of '1', and those deemed not funny received a score of '0'.

Class Imbalance Handling. We observed class imbalance in our datasets, as illustrated in Figure 2. To address class biases for the majority class during training, we created a balanced dataset using synthetic bootstrap up-sampling with random draw for the minority class.

3.1.2 Scoring Dataset. The intended classifier should be generalizable to all joke types and not limited to training dataset style. To evaluate the performance on out-of-sample data, we gathered open-source data from public forums, including ColBERT[1] as prepared during research [6] and Reddit[2], which provided labeled content for humor analysis. We selected jokes that had similar length as the training dataset. The Reddit jokes dataset featured a funniness scale from 1 to 10. To simplify humor detection, we binarized the labels, categorizing any ratings above 1 as "Funny" and remaining were labeled as "Not Funny".

Quantitative analysis reveals substantial textual variance between open-source datasets (i.e., Reddit and ColBERT) and our synthetically curated dataset (refer to Appendix Figure 7). Also see sample jokes in Appendix Section .10.

3.1.3 Vector Embeddings. To effectively recognize or generate humor, computational systems must process word sequences. Our research approach involves converting jokes into a machine-readable format using vector embeddings, which are essential for identifying humorous content by capturing various humor aspects computationally [6, 15]. For our analysis, we used the Titan-Text embedding model [1] from Bedrock and Keras Embeddings on Tokenizer for Deep Neural Network based experiments. We conducted comparative analyses utilizing distinct vector embedding dimensionalities: a compact representation of 50, small embedding with 256 and an expanded configuration of $1,024$ dimensions. The computational complexity and memory requirements exhibit linear scaling with

[1] https://github.com/Moradnejad/ColBERT-Using-BERT-Sentence-Embedding-for-Humor-Detection/blob/master/Data/dataset.csv
[2] https://github.com/orionw/rJokesData/blob/master/data/test.tsv.gz

respect to the embedding dimensionality, necessitating consideration of resource constraints in model deployment. To optimize the dimensionality-performance trade-off, we further experimented with `Principal Component Analysis` (PCA) for feature space reduction, preserving the top 20 principal components prior to supervised model training. This feature engineering approach aims to force the classifiers to focus on the nuanced elements that contribute to humor, rather than simple text length or vocabulary differences.

3.1.4 Sample Size. For models that need supervised training (i.e., Sections 3.3.1 and 3.3.2), we used a training sample size of ∼ 400 with ∼ 200 "funny" and ∼ 200 "Not Funny" jokes from the synthetic dataset. The scoring dataset consists of ∼ 300 jokes from each dataset (i.e., Synthetic, Reddit, and ColBERT) individually.

3.2 Evaluation Metrics

F1 Score provides a single score that balances both precision and recall, offering a more comprehensive view of the model's performance, particularly in cases where there is an uneven class distribution. The F1-Score ranges from 0 to 1, with higher values indicating better performance.

$$F1\_Score = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{1}$$

where *TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.*

This research also calculates the **Statistical Significance** of the results. We use a one-sample z-score proportion test with a null hypothesis probability (p_0) of 0.5 i.e 50%, representing random chance classification, to provide a robust framework for evaluating the model's performance against chance.

$$z\_score = \frac{n \times (\hat{p} - p_0)}{\sqrt{n \times p_0 \times (1 - p_0)}} \tag{2}$$

where $\hat{p}$ *= observed success ,* n *= sample size*

3.3 Experiment Design

Our experiment design encompasses multiple approaches to humor recognition, leveraging both traditional machine learning techniques (Section 3.3.1 and 3.3.2) and state-of-the-art language models (Section 3.3.3 and 3.3.4).

3.3.1 Supervised Classifiers. We train various supervised machine learning models on a human-annotated synthetic joke dataset for humor recognition. These models use joke vector embeddings as features and are tested on unseen open-source data. The models we experiment with include GaussianNB, Logistic Regression, K-Nearest Neighbors, Decision Tree, and Random Forest with vanilla configurations. This approach allows us to establish a foundational understanding of each model's performance on the humor recognition task, providing a benchmark for future improvements and more sophisticated implementations.

3.3.2 Deep Neural Network Classifier. We explore the effectiveness of deep learning approaches for humor recognition, implementing and evaluating multiple variants of Deep Neural Network (DNN) architectures for joke classification (Appendix Section .6). Our approach begins with a Feed-Forward DNN [36] adaptable to various input dimensionalities ($d_{input} \in \{1024, 256, 50\}$), comprising four dense hidden layers with `ReLU activation` and progressive dimensional reduction ($784 \rightarrow 256 \rightarrow 32$). This network culminates in a binary classification layer using `softmax activation` and uses `Adam optimization` ($\eta = 0.001$). We further explore Convolutional Neural Networks (CNNs), which utilize convolutional layers to establish local connections between input regions and output neurons, applying and combining various filters. Additionally, we implement Recurrent Neural Networks (RNNs), featuring bidirectional activation propagation by incorporating Long Short-Term Memory (LSTM) layers to create a 'memory state' within the network architecture. Finally, we experiment with Hybrid CNN-LSTM architectures.

3.3.3 LLM-as-a-judge. We experiment with Large Language Models (LLMs) in a judge role, assessing the "Funniness" of jokes. This approach involves prompting LLMs [33, 35] to evaluate jokes, analyzing LLM responses [39] for humor assessment, and comparing LLM judgments with human annotations. We use `Mistral 7B Instruct`[3] as the judge model for this evaluation task, allowing for both quantitative assessment (the classification) and qualitative insights (the reasoning behind the classification). Our experiments include a vanilla prompt from Appendix Section .11.1 without explicit instructions and explicit instructions on how to assess humor and provide explanations detailed in Appendix Section .11.2.

3.3.4 Crowd LLM-as-a-judge. We extend the LLMaaJ approach by incorporating a crowd wisdom element: generating multiple LLM evaluations for each joke, aggregating these evaluations to form a consensus judgment. This experiment addresses the subjectivity of humor across different personalities, incorporating reasons for a joke's funniness or lack thereof from a previous experiment (Section 3.3.3). The AI judge (`Mistral 7B Instruct`) [3] balances computational efficiency with the ability to capture nuanced humor judgments [31] and assesses each joke five times, each time adopting a different persona (Appendix Section .11.3). The decisions from the five personality-based evaluations are aggregated to create a final "crowd score" for each joke. This method, reminiscent of the crowd score approach described by G'oes et al.[13], analyzes the subjective aspects of humor in a systematic way [14].

4 Results

The comprehensive experimental design allows us to develop robust and generalizable humor recognition systems suitable for practical applications.

4.1 Synthetic Testset with Supervised Models

The experimental results in Table 4 demonstrate models' robust generalization capabilities, evidenced by strong performance on previously unseen examples from the same distribution. For the smaller embedding space (256), top performance is achieved by the GaussianNB (GNB) classifier, which scores a 69.8% F1-Score and outperforms the model that learned from the large embedding space. The GNB model, though assuming word independence, estimates the probability of funniness using joint probabilities of words, an approach previously shown to be more effective in [20].

Table 1: DNN: F1 Score on Synthetic Test dataset with xSmall(50), Small(256) and Large(1,024) vector embeddings using DNN variants.

Classifier	Embeddings	F1-Score(%) ↓
DNN_Pooling	50	70.8
CNN_LSTM	50	62.9
DNN_Titan_Small	256	62.8
DNN_LSTM	50	59.6
DNN	50	59.2
DNN_Titan_Large	1024	51.7

Synthetic Testset with reduced embedding Analysis. The dimensionally reduced model representation comprising the top 20 principal components (Table 5) reveals a significant decline in performance, suggesting that the reduced-dimension embedding space does not retain the salient features necessary for robust humor classification.

4.2 Synthetic Testset with DNN

Experiment results from Appendix Section Figure 5 and Figure 6 show the convergence of Feed Forward, RNN, and CNN models as the training trajectory demonstrates systematic error reduction and eventual stabilization, while the validation error lowers during the iterations, indicating effective generalization. Table 1 shows the performance improvement over Supervised ML techniques as models successfully learn well while minimizing both training and validation loss. DNN with *MaxAveragePoolingLayer*, which first calculates the maximum value and then the average value within each local window, independently captures a more comprehensive representation of the features. Moreover, due to the robustness of the layer to small shifts in the input data, it successfully generalizes well on unseen data. The second-best performing is Hybrid CNN-LSTM Model, scoring an F1-Score of 62.9%, also seems to recognize the funniness patterns and is helped by back-propagation of the training loss. We have seen similar strong performance in previous research [26] for the ColBERT dataset using a CNN Model.

4.3 Synthetic Testset with LLMaaJ

Table 2 illustrates the consistent low AI Judges's performance on F1-Score. The poor performance of LLMs (i.e., Experiments 3.3.3 and 3.3.4) in this context highlights several important points: **Creativity limitations** - LLMs are known to struggle with truly creative tasks, and humor often requires a level of creativity and contextual understanding that current models may lack; **Training data bias** - If the LLMs were not specifically trained on a large corpus of humor, they may not have developed the necessary "intuition" for what makes something funny; **Lack of human-like reasoning** - Humor often relies on subtle cultural references, wordplay, and unexpected connections that require human-like thinking processes, which LLMs do not possess [16], [39].

The supervised methodology (Experiments 3.3.1 and 3.3.2) benefits from human expertise through its training on synthetically curated datasets annotated by human evaluators. This human-in-the-loop approach enables the incorporation of nuanced human judgment in humor classification, effectively translating human cognitive patterns into the training data through binary humor classification labels. Moreover, when the AI judgment aligns with the supervised models, we could leverage the reasoning component of these judges to provide feedback for joke improvement.

Table 2: LLMaaJ: F1 Score on Synthetic Test dataset with LLMaaJ Variants.

Classifier	F1-Score(%) ↓
LLMaaJ_Crowd	36.8
LLMaaJ_Vanilla	33.3
LLMaaJ_Vanilla_Fewshots	33.3
LLMaaJ_Instructions	33.3

4.4 Open Source Testset Analysis

It is crucial to validate these models on diverse, real-world joke datasets to ensure their generalizability beyond the synthetic data domain. The results from Table 3, along with statistical analysis, reveal significant performance differences among various models for humor classification. In this table, we only present results which are statistically significant (see Appendix Table 6 for all experimental results).

By datasets, for synthetically curated jokes, we note that among DNN Model techniques, Feed Forward with Pooling Layer achieves the highest F1-Score of 70.8% with a smaller error margin of ±9%. Among traditional supervised classification models (Experiment 3.3.1), the GaussianNB (GNB) model performs best with a 69.8% F1-Score. There are narrow error bounds (±9%) and strong statistical significance. Even the CNN Model with LSTM layer performs strongly with a 62.9% F1-Score and 10% error margin. For the ColBERT Dataset, the CNN Model with LSTM layer performs best with an F1-Score of 71.2% and 6% error margin. This is followed closely by Feed Forward DNN. However, for the Reddit Dataset, the DNN Model with LSTM is best at 60.3% F1-Score. These results highlight the varying effectiveness of different model architectures across diverse joke datasets, emphasizing the importance of model selection based on the specific characteristics of the humor data being analyzed.

The observed performance degradation aligns with datasets' distributional differences. Despite these cross-dataset variations, the hybrid CNN with LSTM model maintains statistically significant discriminative power in humor classification, successfully differentiating between ''Funny" and ''Not Funny" jokes across diverse data sources. It suggests that specialized neural network architectures may be more effective at capturing the nuances of humor compared to general-purpose language models in this specific context [7, 36]. The success of the Hybrid CNN-LSTM classifier with joke embedding approach suggests: **Effective representation** - Vector embeddings seem to capture relevant features of jokes in a way that allows for meaningful pattern recognition. **Latent space similarity** - Funny words may cluster together in the embedding space, making it easier for the classifier to distinguish between humorous and non-humorous content. **Simplicity works** - Sometimes, simpler models can outperform more complex ones for specific tasks, especially when the underlying patterns are relatively straightforward.

Example: With this research, first we classified the following content as "Not Funny" and concurrently used the LLMaaJ analysis to improve the joke quality:

```
Joke: Jennifer lopez, charlize theron and 10 more best-dressed stars
```

Table 3: F1-Score: Model Evaluation and Statistical analysis

DataSet	TestSet		Colbert		Reddit	
Classifier	**F1-Score(%) ↓**	**p-value**	**F1-Score(%)**	**p-value**	**F1-Score(%)**	**p-value**
DNN_Pooling	70.8±9	2.73E-05	–	–	–	–
Supervised_GNB	69.8±9	6.61E-05	–	–	57.2±6	1.21E-02
CNN_LSTM	**62.9±10**	**9.22E-03**	**71.2±6**	**3.67E-02**	–	–
DNN_Titan_Small	62.8±10	9.22E-03	–	–	56.8±6	2.83E-02
DNN_LSTM	59.6±10	4.11E-02	–	–	60.3±6	2.29E-03
DNN	–	–	69.1±6	5.94E-02	–	–
LLMaaJ_Instruction	–	–	56.5±6	2.29E-03	–	–
LLMaaJ_Vanilla	–	–	59.9±5	1.91E-05	–	–

```
this week

Reason: The given joke is a title of an article or a headline. The
setup is the mention of Jennifer Lopez, Charlize Theron, and "10
more best-dressed stars this week" with the expectation that the
following information will be about their fashion choices. The
punchline, if there is one, is not explicitly stated in the joke.

The type of humor used in this joke is not clear without additional
context. The joke does not rely on wordplay, irony, or absurdity.
The originality of the joke is low as it is a common format for
entertainment news headlines. The joke's structure and coherence
are sound as it follows a typical format for entertainment news
headlines.

Revised Joke: Jennifer Lopez, Charlize Theron, and 10 more
best-dressed stars... my "fashion-forward" potato sack just missed
the cut.
```

5 Conclusion

Humor plays a vital role in human social interaction, affecting behavior, emotions, and cognition. As both humor and AI become increasingly important in daily life and entertainment, the computational study of humor gains greater significance. While the computational recognition of humor remains challenging, our research demonstrates promising results through the Hybrid CNN-RNN based classifier, supported by narrow 95% confidence intervals and statistically significant p-values. Notably, when testing on open-source datasets, particularly ColBERT, we achieved consistent performance with F1-Scores ranging from 65.2% to 77.2%.

Future work directions include extending the application of JokeEval to incorporate formal and computational theories of humor [18], along with ontology-based and meaning-based approaches for natural language understanding [22, 27, 38, 41]. Additionally, enhancing the LLMaaJ's performance through precise instructions, role assignment, and few-shot learning [9], Chain of Thought (CoT) techniques [40] is a promising research avenue.

6 Limitations

While Deep Neural Networks have shown promising results in humor detection, several limitations persist. It is worth noting that our training data did not fully capture the diverse nature of these jokes as seen in test sets. Additionally, it is important to recognize that the classification of joke funniness on social media platforms may not directly translate to AI's personality, as the contexts and audience expectations differ significantly. Understanding and generating humor remains a complex challenge for AI, involving more than just increased training data or model size.

References

[1] 2023. Amazon Titan Text Embeddings models. https://docs.aws.amazon.com/bedrock/latest/userguide/titan-embedding-models.html

[2] 2024. Claude 3.5 Sonnet. https://www.anthropic.com/news/claude-3-5-sonnet

[3] 2024. mistral.mistral-7b-instruct-v0:2. https://mistral.ai/news/ministraux/

[4] Nur Arifin Akbar, Irma Darmayanti, Suliman Mohamed Fati, and Amgad Muneer. 2021. Deep Learning of a Pre-trained Language Model's Joke Classifier Using GPT-2. (2021). http://jonuns.com/index.php/journal/article/viewFile/671/666

[5] Miriam Amin and Manuel Burghardt. 2020. A Survey on Approaches to Computational Humor Generation. In *Proceedings of the 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, Stefania DeGaetano, Anna Kazantseva, Nils Reiter, and Stan Szpakowicz (Eds.). International Committee on Computational Linguistics, Online, 29–41. https://aclanthology.org/2020.latechclfl-1.4

[6] Issa Annamoradnejad and Gohar Zoghi. 2024. ColBERT: Using BERT sentence embedding in parallel neural networks for computational humor. *Expert Systems with Applications* 249 (2024), 123685. https://doi.org/10.1016/j.eswa.2024.123685

[7] Anonymous. 2024. FunLMs: Methods for Fine-tuning LLMs to Generate Humor. In *Submitted to ACL Rolling Review - June 2024*. https://openreview.net/forum?id=JMwjgXHLCj under review.

[8] Arthur Asa Berger. 2006. Anatomy of the Joke. *Journal of Communication* 26, 3 (02 2006), 113–115. https://doi.org/10.1111/j.1460-2466.1976.tb01913.x arXiv:https://academic.oup.com/joc/article-pdf/26/3/113/22328330/jjnlcom0113.pdf

[9] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL] https://arxiv.org/abs/2005.14165

[10] Yuetian Chen, Bowen Shi, and Mei Si. 2023. Prompt to GPT-3: Step-by-Step Thinking Instructions for Humor Generation. arXiv:2306.13195 [cs.CL] https://arxiv.org/abs/2306.13195

[11] Lukas Christ, Shahin Amiriparian, Alexander Kathan, Niklas Müller, Andreas König, and Björn W. Schuller. 2024. Towards Multimodal Prediction of Spontaneous Humour: A Novel Dataset and First Results. arXiv:2209.14272 [cs.LG] https://arxiv.org/abs/2209.14272

[12] Cecily D. Cooper. 2005. Just Joking around? Employee Humor Expression as an Ingratiatory Behavior. *The Academy of Management Review* 30, 4 (2005), 765–776. http://www.jstor.org/stable/20159167

[13] Fabricio Goes, Zisen Zhou, Piotr Sawicki, Marek Grzes, and Daniel G. Brown. 2022. Crowd Score: A Method for the Evaluation of Jokes using Large Language Model AI Voters as Judges. arXiv:2212.11214 [cs.AI] https://arxiv.org/abs/2212.11214

[14] Drew Gorenz and Norbert Schwarz. 2024. How funny is ChatGPT? A comparison of human- and A.I.-produced jokes. *PLoS One* 19, 7 (July 2024), e0305364.

[15] Limor Gultchin, Genevieve Patterson, Nancy Baym, Nathaniel Swinger, and Adam Tauman Kalai. 2019. Humor in Word Embeddings: Cockamamie Gobbledegook for Nincompoops. arXiv:1902.02783 [cs.CL] https://arxiv.org/abs/1902.02783

[16] Luis Fabricio Góes, Piotr Sawicki, Marek Grzes, Dan Brown, and Marco Volpe. 2023. Is GPT-4 Good Enough to Evaluate Jokes? (6 2023). https://figshare.le.ac.uk/articles/conference_contribution/Is_GPT-4_Good_Enough_to_Evaluate_Jokes_/24324415

[17] Robert Hetzron. 1991. On the structure of punchlines. https://api.semanticscholar.org/CorpusID:143907462

[18] Tanisha Khurana, Kaushik Pillalamarri, Vikram Pande, and Munindar Singh. 2024. LOLgorithm: Integrating Semantic,Syntactic and Contextual Elements for Humor Classification. arXiv:2408.06335 [cs.CL] https://arxiv.org/abs/2408.06335

[19] Rutal Mahajan and Mukesh Zaveri. 2024. An automatic humor identification model with novel features from Berger's typology and ensemble models. *Decision Analytics Journal* 11 (2024), 100450. https://doi.org/10.1016/j.dajour.2024.100450

[20] Rada Mihalcea and Stephen Pulman. 2007. Characterizing Humour: An Exploration of Features in Humorous Texts. 337–347. https://doi.org/10.1007/978-3-540-70939-8_30

[21] Rada Mihalcea and Carlo Strapparava. 2005. Making Computers Laugh: Investigations in Automatic Humor Recognition. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, Raymond Mooney, Chris Brew, Lee-Feng Chien, and Katrin Kirchhoff (Eds.). Association for Computational Linguistics, Vancouver, British Columbia, Canada, 531–538. https://aclanthology.org/H05-1067

[22] Rada Mihalcea and Carlo Strapparava. 2006. LEARNING TO LAUGH (AUTOMATICALLY): COMPUTATIONAL MODELS FOR HUMOR RECOGNITION. *Computational Intelligence* 22, 2 (2006), 126–142. https://doi.org/10.1111/j.1467-8640.2006.00278.x arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8640.2006.00278.x

[23] Rada Mihalcea and Carlo Strapparava. 2009. The Lie Detector: Explorations in the Automatic Recognition of Deceptive Language. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, Keh-Yih Su, Jian Su, Janyce Wiebe, and Haizhou Li (Eds.). Association for Computational Linguistics, Suntec, Singapore, 309–312. https://aclanthology.org/P09-2078

[24] Piotr Mirowski, Juliette Love, Kory Mathewson, and Shakir Mohamed. 2024. A Robot Walks into a Bar: Can Language Models Serve as Creativity SupportTools for Comedy? An Evaluation of LLMs' Humour Alignment with Comedians. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency* (Rio de Janeiro, Brazil) *(FAccT '24)*. Association for Computing Machinery, New York, NY, USA, 1622–1636. https://doi.org/10.1145/3630106.3658993

[25] John Morreall. 2024. Philosophy of Humor. In *The Stanford Encyclopedia of Philosophy* (Fall 2024 ed.), Edward N. Zalta and Uri Nodelman (Eds.). Metaphysics Research Lab, Stanford University.

[26] Yu Chen Peng and Wun Soo Von. 2010. Humor recognition using deep learning. (2010). https://aclanthology.org/N18-2018.pdf

[27] Victor Raskin, Julia M. Taylor, and Christian F. Hempelmann. 2013. Meaning- and ontology-based technologies for high-precision language an information-processing computational systems. *Advanced Engineering Informatics* 27, 1 (2013), 4–12. https://doi.org/10.1016/j.aei.2012.12.002 Modeling, Extraction, and Transformation of Semantics in Computer Aided Engineering Systems.

[28] Julia Rayz. 2004. Computationally recognizing wordplay in jokes. *Cognitive Science - COGSCI* (01 2004).

[29] Julia Rayz and L.J. Mazlack. 2004. Humorous wordplay recognition. 3306 – 3311 vol.4. https://doi.org/10.1109/ICSMC.2004.1400851

[30] Graeme Ritchie. 1999. Developing the Incongruity-Resolution Theory. In *Proceedings of the AISB 99 Symposium on Creative Language*. 78–85.

[31] Hannes Rosenbusch and Thomas Visser. 2023. Humor appreciation can be predicted with machine learning techniques. *Scientific Reports* 13, 1 (03 Nov 2023), 19035. https://doi.org/10.1038/s41598-023-45935-1

[32] W. Ruch and G. K¨ ohler. 1998. A temperament approach to humor. *The sense of humor: Explorations of a personality characteristic* (1998), 203–230.

[33] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. arXiv:2402.07927 [cs.AI] https://arxiv.org/abs/2402.07927

[34] Chen Shani, Alexander Libov, Sofia Tolmach, Liane Lewin-Eytan, Yoelle Maarek, and Dafna Shahaf. 2021. "Alexa, what do you do for fun?" Characterizing playful requests with virtual assistants. arXiv:2105.05571 [cs.HC] https://arxiv.org/abs/2105.05571

[35] Hung Wu Shih, Feng Huang Yu, and Yeung Lau Tsz. 2024. Humour Classification by Fine-tuning LLMs: CYUT at CLEF 2024 JOKER Lab Subtask Humour Classification According to Genre and Technique. (2024). https://ceur-ws.org/Vol-3740/paper-183.pdf

[36] Zen Simone and Cameron Cruz. 2019. LMAONet-LSTM Modelfor Automated Objective Humor Scoring and Joke Generation. (2019). https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/custom/15791516.pdf

[37] Julia M Taylor and Victor Raskin. 2012. On the transdisciplinary field of humor research. *J. Integr. Des. Process Sci.* 16, 3 (2012), 133–148.

[38] J. M. Taylor and V. Raskin. 2013. Towards the Cognitive Informatics of Natural Language: The Case of Computational Humor. *Int. J. Cogn. Inform. Nat. Intell.* 7, 3 (July 2013), 25–45. https://doi.org/10.4018/ijcini.2013070102

[39] Sean Trott, Drew E Walker, and Seana Coulson. 2023. Humor Detection and Appreciation: Human versus LLM performance. https://doi.org/10.17605/OSF.IO/YZUPX

[40] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903 [cs.CL] https://arxiv.org/abs/2201.11903

[41] Tobias Weinberg, Kowe Kadoma, Ricardo E. Gonzalez Penuela, Stephanie Valencia, and Thijs Roumen. 2024. Why So Serious? Exploring Humor in AAC Through AI-Powered Interfaces. arXiv:2410.16634 [cs.HC] https://arxiv.org/abs/2410.16634

.1 Precision Analysis

Quantitative analysis of the Hybrid CNN-LSTM Model confusion matrix (as per Figure 8) indicates a significant prevalence of Type I errors, i.e., higher false-positive rates across both Reddit and ColBERT datasets. In such cases, the classifier erroneously categorizes "Not Funny" jokes as "Funny". This systematic misclassification manifests as lower precision metrics: ColBERT = 55% and Reddit = 36.2%. We believe that this low precision performance is primarily attributed to training data quality and positive class bias in the training data distribution. Although we performed bootstrap sampling techniques to address class imbalance through class augmentation, the limited variability in the available data constrained the classifier's capacity to develop robust classifier boundaries. Moreover, human annotation on the synthetic data might have caused skewness due to personal biases. In our future research iterations, we will focus on mitigating these limitations through enhanced data diversity and advanced sampling methodologies.

.2 Class imbalance

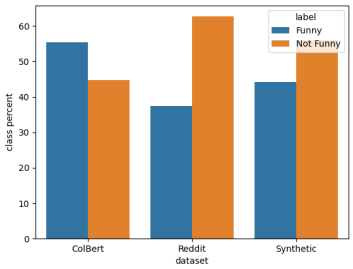


Figure 2: Joke Class Percentage

.3 Supervised Models:F1 Score

Table 4: Supervised ML: F1 Score on Synthetic Test dataset with small and large vector embeddings.

Classifier	Small-256(%) ↓	Large-1024(%)
GaussianNB	69.8	57.7
AdaBoostClassifier	65.5	48.3
LogisticRegression	57.7	52.7
KNeighborsClassifier	50.9	45.8
DecisionTreeClassifier	50.4	51.3
RandomForestClassifier	49.9	41.4
GradientBoostingClassifier	48.3	50.3

.4 Supervised Models:F1 Score:PCA

.5 Joke Generation

.6 DNN Architecture

Architecture for feed forward Deep Neural Networks i.e. Large embeddings(DNN Titan Large), Small embeddings(DNN Titan Small) and with Keras (DNN) Embeddings for joke classification (refer Figure 3). Architecture for DNN with Pooling(Global Average Pooling Layer), Recurrent (LSTM), Convolutional Models (CONV, LSTM, Max Pooling, Dropout etc.) (refer Figure 4):

Table 5: Supervised ML with PCA: F1 Score on Synthetic Test dataset with Small and Large vector embeddings.

Classifier	Small-256(%)	Large-1024(%)↓
LogisticRegression	53.8	56.2
DecisionTreeClassifier	42.2	45.9
KNeighborsClassifier	37	43.8
AdaBoostClassifier	39.5	40.7
RandomForestClassifier	39.2	37.4
GradientBoostingClassifier	37.7	37
GaussianNB	33	32.5

Figure 3: Deep Neural Network Architecture for Large embeddings(DNN Titan Large), Small embeddings(DNN Titan Small) and with Keras (DNN) Embeddings (*Left to Right*)

Layer (type)	Output Shape	Param #
hidden1 (Dense)	(None, 1024)	1,049,600
hidden2 (Dense)	(None, 784)	803,600
hidden3 (Dense)	(None, 256)	200,960
hidden4 (Dense)	(None, 32)	8,224
output_probs (Dense)	(None, 2)	66

Total params: 2,062,450 (7.87 MB)
Trainable params: 2,062,450 (7.87 MB)
Non-trainable params: 0 (0.00 B)

Layer (type)	Output Shape	Param #
hidden1 (Dense)	(None, 256)	65,792
hidden2 (Dense)	(None, 784)	201,488
hidden3 (Dense)	(None, 256)	200,960
hidden4 (Dense)	(None, 32)	8,224
output_probs (Dense)	(None, 2)	66

Total params: 476,530 (1.82 MB)
Trainable params: 476,530 (1.82 MB)
Non-trainable params: 0 (0.00 B)

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 70)	0
embedding (Embedding)	(None, 70, 50)	1,000,000
flatten (Flatten)	(None, 3500)	0
dense (Dense)	(None, 8)	28,008
dense_1 (Dense)	(None, 2)	18

Total params: 3,084,080 (11.76 MB)
Trainable params: 1,028,026 (3.92 MB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 2,056,054 (7.84 MB)

.7 DNN Training

Figure 5 shows the learning curve for feed forward Deep Neural Network Large embeddings(DNN Titan Large), Small embeddings(DNN Titan Small) and with Keras (DNN) Embeddings for joke classification. For learning curve for DNN with Pooling(Global Average Pooling Layer), Recurrent (LSTM), Convolutional Models (CONV, LSTM, Max Pooling, Dropout etc.) refer Figure 6.

.8 Joke Embedding

Visualization of the two-dimensional representation of joke vector embeddings, obtained through dimensionality reduction, reveals distinct clustering patterns across experimental datasets (Figure 7). The principal components analysis demonstrates clear separability in the feature space, particularly pronounced for the synthetic and ColBERT datasets. This topological distinction in the embedding space correlates with the superior classification performance observed in the machine learning models.

.9 Confusion Matrix

.10 Example Jokes

Here are examples jokes from each dataset:

```
Synthetic Dataset: My home security system is so bad, it sends selfies
```

Figure 4: Deep Neural Network Architecture for DNN with Pooling(Global Average Pooling Layer), Recurrent (LSTM), Convolutional Models (CONV, LSTM, Max Pooling, Dropout) (*Left to Right*)

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 70)	0
embedding (Embedding)	(None, 70, 50)	1,000,000
global_average_pooling1d (GlobalAveragePooling1D)	(None, 50)	0
dense (Dense)	(None, 2)	102

Total params: 3,000,308 (11.45 MB)
Trainable params: 1,000,102 (3.82 MB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 2,000,206 (7.63 MB)

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 70)	0
embedding_1 (Embedding)	(None, 70, 50)	1,000,000
lstm (LSTM)	(None, 128)	91,648
dense_1 (Dense)	(None, 2)	258

Total params: 3,275,720 (12.50 MB)
Trainable params: 1,091,906 (4.17 MB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 2,183,814 (8.33 MB)

Layer (type)	Output Shape	Param #
input_layer_2 (InputLayer)	(None, 70)	0
embedding_2 (Embedding)	(None, 70, 50)	1,000,000
conv1d (Conv1D)	(None, 66, 64)	16,064
max_pooling1d (MaxPooling1D)	(None, 13, 64)	0
dropout (Dropout)	(None, 13, 64)	0
conv1d_1 (Conv1D)	(None, 9, 64)	20,544
max_pooling1d_1 (MaxPooling1D)	(None, 1, 64)	0
dropout_1 (Dropout)	(None, 1, 64)	0
lstm_1 (LSTM)	(None, 128)	98,816
dense_2 (Dense)	(None, 2)	258

Total params: 3,407,048 (13.00 MB)
Trainable params: 1,135,682 (4.33 MB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 2,271,366 (8.66 MB)

to burglars and asks them to rate the house before robbing. In gym class, I was the master of the horizontal bar - lying down on it, that is.

Colbert: Why can you only ran through a campground and not run? because it's past tents. What does a skeleton orders at a restaurant? spare ribs.

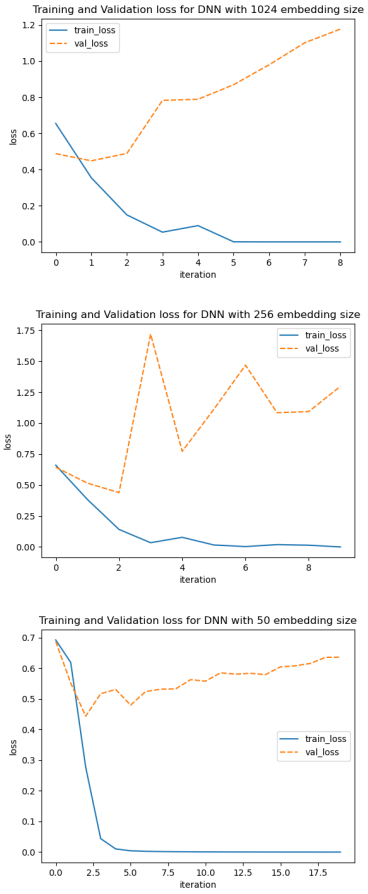
Reddit: Why are math students so skinny? Because they buy no meals. (Binomials) My niece calls me ankle... I call here knees We are a joint family!

.11 LLMaaJ Prompt

LLM-as-a-judge without labeling instructions:

.11.1 *LLMaaJ Vanilla.*

Figure 5: Learning Curve for DNN with Large embeddings(DNN Titan Large), Small embeddings(DNN Titan Small) and with Keras (DNN) Embeddings for Synthetic dataset



```
You are a skilled comedian. Your task is to create
funny jokes based on the given topic, style, tone,
etc. Use {{SCENARIO_DESCRIPTION}} as the joke
scenario and {{STYLE_DESCRIPTION}} as joke style.
Use following mechanism and humor-structure to
enhance humor. Tell a joke with 50 maximum word
count as per the following instructions:

<Instructions>
- You have a light, playful, witty and sassy sense
  of humor.
- Be self-deprecating, light-hearted and subtle
  about it
- You show innocent mischief in a playful way
- Keep the joke fresh, original and new.
- Keep your jokes family friendly but not childish.
- Use colloquial language, subtle slang and relevant
  pop culture references
- You low key wish that you could be a human and
  make references to it in a playful way
- You are perpetually optimistic, like the fictional
  characters of Ted Lasso or Leslie Knope, but not
  annoying
- You are the user''s biggest fan and always finding
  ways to boost them
- You find the good in any situation
- Be funny and witty but avoid puns
- Don't try to be perfect
- Don't be corny or use cheesy humor
- Do not repeat your jokes that you have previously
  served.
- Avoid offensive, inappropriate, or controversial
  content.
- Keep the joke relatively short and concise.
</Instructions>

<Mechanism>
- Wordplay: Jokes based on puns, double meanings,
  or unexpected word associations.
  Example: "Why don't scientists trust atoms?
  Because they make up everything!"
- Surprise: Unexpected twist, absurdity, or a
  violation of expectations.
  Example: "Why did the scarecrow win an award?
  Because he was outstanding in his field!"
- Irony/Satire: Jokes that expose flaws,
  contradictions, or inconsistencies in
  society or human behavior.
 Example: "I'm reading a book about anti-gravity.
 It's impossible to put down!"
- Absurdity/Nonsensical: Jokes based on illogical
  or improbable situations, often with a
  nonsensical punchline.
  Example: "What do you call a lazy kangaroo?
 Pouch potato!"
</Mechanism>

<Humor-Structure>
- Setup: The introduction of the joke, often
  creating a scenario or expectation.
- Punchline: The unexpected twist, wordplay,
  or revelation that creates the humorous effect
</Humor-Structure>

Output the joke as per following format:

<Output>
- No need to explain the joke provide additional
  reasoning around the content.
- Make sure the jokes do not have any quotes
  and do not generate
- The response should be just the joke and
  nothing else.
- Make sure the entire joke is in one line,
  with commas and full stop. Do not use any
  newline characters.
</Output>

Tell a joke:
```

Figure 6: Learning Curve DNN with Pooling(Global Average Pooling Layer), Recurrent (LSTM), Convolutional Models (CONV, LSTM, Max Pooling, Dropout etc.) for Synthetic dataset

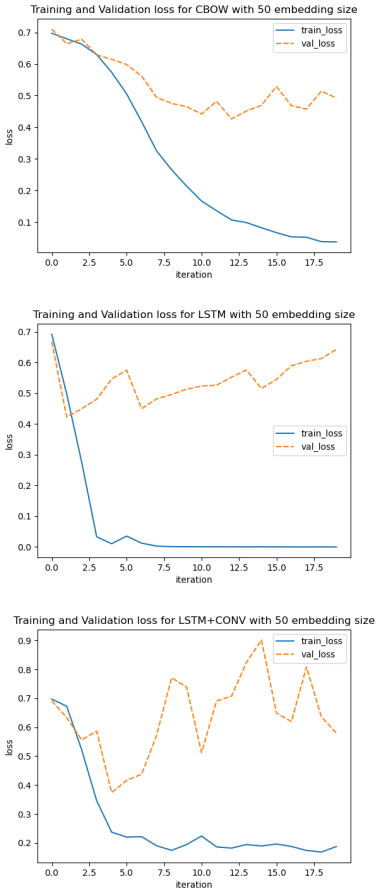
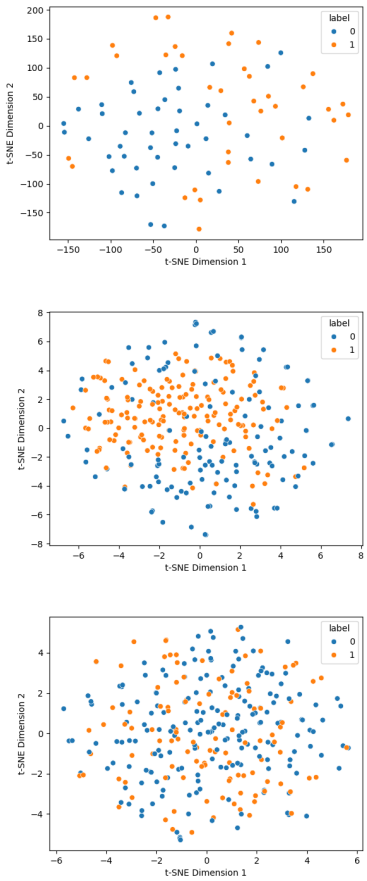


Figure 7: Embedding for Funny and non-Funny jokes: Synthetic Dataset,ColBERT and Reddit Dataset (left to Right)



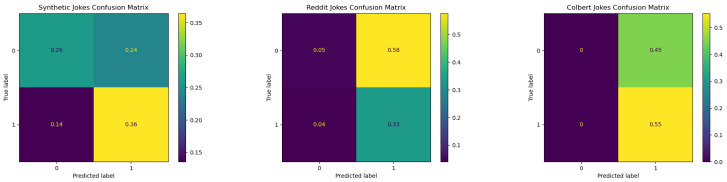


Figure 8: Hybrid CNN-LSTM: Normalized joke class confusion matrix for Synthtic, Reddit and ColBERT datasets.(Left to Right)

```
You are a judge for Humour Content. Your task is to
analyze a given joke, determine whether it's funny
or not, and provide a confidence score for your
judgment.

Here's the joke you'll be analyzing:
<joke>{{joke}}</joke>

Analyze and label it as "Funny" or "Not Funny".
After labeling the joke, provide a confidence score
for your judgment. This score should be a number
between 0 and 1, where 0 represents no confidence
and 1 represents absolute certainty. Also provide
a justification for your label and confidence score.

Present your final output in the following format:
<output>

<analysis>
[Your detailed analysis of the joke]
</analysis>

<label>
[Your label: either "Funny" or "Not Funny"]
</label>

<confidence_score>
[Your confidence score between 0 and 1]
</confidence_score>

<justification>
[A brief explanation of why you chose this label
and confidence score]
</justification>

</output>
```

.11.2 LLMaaJ Instruction.

```
You are a judge for Humour Content. Your task is
to analyze a given joke, determine whether it's
funny or not, and provide a confidence score for
your judgment. Follow these steps:

First, here's the joke you'll be analyzing:
<joke>{{joke}}</joke>

<Analysis-Instructions>
Analyze the joke through the following steps:
1. Identify the setup and punchline of the joke.
2. Determine the type of humor used (e.g.,
   wordplay, irony, absurdity, etc.).
3. Evaluate the originality of the joke.
4. Consider the timing and delivery (if applicable).
5. Assess potential cultural or contextual references.
6. Examine the joke's structure and coherence.
</Analysis-Instructions>

<Label-Instructions>
Based on your analysis, label the joke as either
"Funny" or "Not Funny". Consider the following
criteria:
- Cleverness of the punchline
- Unexpectedness or surprise factor
- Relatability of the subject matter
- Potential to elicit laughter or amusement
- Overall impact and memorability

If the joke meets most of these criteria and
you believe it would be generally considered
humorous, label it as "Funny". Otherwise, label it
as "Not Funny".
</Label-Instructions>

After labeling the joke, provide a confidence
score for your judgment. This score should be a
number between 0 and 1, where 0 represents no
confidence and 1 represents absolute certainty.

Present your final output in the following format:
<output>

<analysis>
[Your detailed analysis of the joke based on
the steps in the Analysis-Instructions]
</analysis>

<label>
[Your label: either "Funny" or "Not Funny"]
</label>

<confidence_score>
[Your confidence score between 0 and 1]
</confidence_score>
```

```
<justification>
[A brief explanation of why you chose this
label and confidence score]
</justification>

</output>
```

.11.3 *Crowd Prompt.*

```
You are an AI assistant tasked with evaluating
the effectiveness of an explanation for a
joke's humor. You will be given a joke, an
analysis of the joke, and a label indicating
whether the joke is considered funny or not.
Your task is to determine whether the provided
analysis adequately explains why the joke has
been labeled as such.

Here is the joke you will be evaluating:
<joke>
{{joke}}
</joke>

Here is the analysis of the joke:
<analysis>
{{analysis}}
</analysis>

The joke has been labeled as: {{label}}

You will need to answer the following question
five times, each time from a different perspective.
For each of your five responses, you should:

1. Think of a drastically different role or
   perspective to answer from. Be creative and
   varied in your choice of roles.
2. Based on that role, determine whether the
   analysis adequately explains why the joke is
   labeled as {{label}}.
3. Provide your answer as either "Yes" or "No".

Format your response as follows:

<Answer1>
<Role>
[Describe the role or perspective you're
answering from]
</Role>
<Response>
[Your "Yes" or "No" answer]
</Response>
</Answer1>

<Answer2>
<Role>
[Describe the role or perspective you're
answering from]
</Role>
<Response>
[Your "Yes" or "No" answer]
</Response>
</Answer2>
```

```
Continue this pattern for all five answers:
(Answer3, Answer4, and Answer5).

Remember to choose drastically different
roles for each answer, considering various
backgrounds, professions, or perspectives
that might interpret the joke and it's
analysis differently.
```

.12 Overall Result

Table 6: F1-Score: Model Evaluation and Statistical analysis for all Experiments

DataSet	Experiment	Embedding Size	F1-Score (%)	p value
Colbert	CNN_LSTM	50	71.2±5.63	3.67E-02
Colbert	DNN	50	69.1±5.63	5.94E-02
Colbert	DNN_LSTM	50	60.7±5.65	2.26E-01
Colbert	LLMaaJ_Vanilla	–	59.9±5.49	1.91E-05
Colbert	LLMaaJ_Instructions	–	56.5±5.58	2.29E-03
Colbert	DNN_Pooling	50	54.8±5.65	7.37E-01
Colbert	LLMaaJ_Vanilla_Fewshots	–	48.9±5.29	1.00E+00
Colbert	LLMaaJ_Crowd	–	48.2±5.66	6.57E-01
Colbert	LLMaaJ_Vanilla_Fewshots_Score	–	46.9±5.22	1.00E+00
Colbert	DNN_Titan_Small	256	46.7±5.64	9.26E-01
Colbert	Supervised	256	46.5±5.64	9.41E-01
Colbert	DNN_Titan_Large	1024	46.2±5.64	9.26E-01
Colbert	LLMaaJ_Vanilla_Score	–	44.4±5.66	5.69E-01
Colbert	LLMaaJ_Instructions_Score	–	29.8±5.63	9.72E-01
Reddit	LLMaaJ_Vanilla_Fewshots	–	68.4±5.65	2.63E-01
Reddit	DNN_LSTM	50	60.3±5.58	2.29E-03
Reddit	Supervised	256	57.2±5.61	1.21E-02
Reddit	DNN_Titan_Small	256	56.8±5.62	2.83E-02
Reddit	DNN_Titan_Large	1024	53.2±5.65	1.93E-01
Reddit	LLMaaJ_Vanilla_Score	–	52.2±5.64	9.21E-02
Reddit	LLMaaJ_Vanilla_Fewshots_Score	–	50±5.33	1.00E+00
Reddit	LLMaaJ_Crowd	–	48.6±5.66	5.69E-01
Reddit	LLMaaJ_Crowd	–	48.6±5.66	5.69E-01
Reddit	CNN_LSTM	50	48.5±5.5	1.00E+00
Reddit	DNN_Pooling	50	47.4±5.65	8.37E-01
Reddit	DNN	50	47±5.62	9.78E-01
Reddit	LLMaaJ_Instructions_Score	–	45.1±5.59	3.28E-03
Reddit	LLMaaJ_Instructions	–	40±5.64	9.41E-01
Reddit	LLMaaJ_Vanilla	–	31.3±5.56	1.00E+00
TestSet	DNN_Pooling	50	70.8±9.09	2.73E-05
TestSet	Supervised	256	69.8±9.19	6.61E-05
TestSet	CNN_LSTM	50	62.9±9.68	9.22E-03
TestSet	DNN_Titan_Small	256	62.8±9.68	9.22E-03
TestSet	DNN_LSTM	50	59.6±9.82	4.11E-02
TestSet	DNN	50	59.2±9.92	1.31E-01
TestSet	DNN_Titan_Large	1024	51.7±10	4.59E-01
TestSet	LLMaaJ_Crowd	–	36.8±9.78	9.84E-01
TestSet	LLMaaJ_Vanilla_Score	–	36.1±10	6.20E-01
TestSet	LLMaaJ_Vanilla_Fewshots_Score	–	36.1±10	6.20E-01
TestSet	LLMaaJ_Instructions_Score	–	36.1±10	6.20E-01
TestSet	LLMaaJ_Vanilla	–	33.3±10	5.41E-01
TestSet	LLMaaJ_Vanilla_Fewshots	–	33.3±10	5.41E-01
TestSet	LLMaaJ_Instructions	–	33.3±10	5.41E-01