PunDerstand @ CLEF JOKER 2024: Who's Laughing Now? Humor Classification by Genre & Technique

Notebook for the JOKER Lab at CLEF 2024

Ryan Rony Dsilva*,†, Nidhi Bhardwaj†

Purdue University, West Lafayette, IN, USA

Abstract

Humor is subject to individual interpretation, with each person perceiving it differently. Given that humor itself is subjective, this work explores classification of humor by genre and technique through three approaches: manual guided annotation, multi-class classification using BERT-based models with and without sampling, and prompting with large language models. Our experiments revealed insights into the performance of different models and approaches on the humor classification task and opens up further discussions on using guidelines from the annotation to aid large language models.

Keywords

humor classification, BERT, large language models, humor theory, guided annotation

1. Introduction

Humor can be perceived differently based on an individual's perspective [1]. What one person considers humorous, another may not, and even an individual's sense of humor can change depending on their mood or recent experiences. This makes the task of identifying the specific type of humor even more challenging. The CLEF JOKER 2024 track [2, 3] proposed 3 tasks, but we focus on Task 2: Humor classification according to genre and technique [4] in which the task was to identify the particular type of humor found in the text from among - irony, sarcasm, exaggeration, incongruity-absurdity, self-deprecating and wit-surprise. Our approach to this task involves an approach mimicking the evolution of the field from manual annotation using humor theories as guides to using powerful large language models through prompting. The idea is to discover how various approaches that evolved with time perform on the same underlying problem of subjectivity in humor. We present the results along with analysis of the patterns found during our experiments to aid further research.

2. Methodology

For this task, the dataset contained manually annotated examples from the JOKER 2023 corpus [5] as well as new data.

2.1. Guided Annotation

To classify the type of humor present in sentences, we implemented a guided annotation process. This method involved developing a comprehensive codebook that provided explicit guidelines for categorizing sentences into predefined humor types. To minimize bias arising from preconceived notions of humor, we assigned pseudo names to the categories. This anonymization aimed to ensure that annotators based their classification decisions solely on the structural and contextual cues detailed

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

[*] Corresponding author.

[†] These authors contributed equally.

ryan.rony.dsilva@gmail.com (R. R. Dsilva); nidhi.bhardwaj012@gmail.com (N. Bhardwaj)
https://www.ryandsilva.dev (R. R. Dsilva)
0009-0008-4158-4504 (R. R. Dsilva); 0009-0000-0294-7677 (N. Bhardwaj)

in the codebook, rather than on any prior subjective understanding of the humor types. Two annotators were tasked with categorizing humor based on guidelines outlined in the provided codebook. Sentences where both annotators agreed on the humor category were considered final and included for submission, while those with disagreement were excluded. Upon reaching consensus, a total of 350 sentences were submitted as the final annotated dataset. The codebook outlined specific characteristics and markers for each humor type. The instructions were derived from the definitions of each category and the patterns observed in the training dataset. The detailed codebook used in this process is included in the appendix.

Construction of the Codebook

To classify wit in sentences, we adopted definitions from [6], which describe wit as involving an unexpected twist or element that generates humor. In the training dataset, sentences containing wit often exhibited patterns like the use of words with multiple meanings or homophones. These linguistic features were incorporated to facilitate easier identification of wit.

For categorizing sentences as incongruous or absurd, the training dataset revealed a common bipartite structure: the first part typically posed a question, followed by an unexpected or unrealistic answer. Absurd humor was detected by identifying nonsensical situations that elicited humor [7]. Any humorous characteristic that appeared illogical or unrealistic was classified under this category.

Self-deprecation was identified by structural cues indicating that a sentence negatively addressed oneself [8]. The humor in self-deprecating jokes arises from highlighting one's weaknesses and flaws in an embarrassing yet unexpected manner, as it is uncommon for individuals to discuss their shortcomings openly [9].

To identify exaggeration, we focused on detecting hyperbolic terms [10] within sentences that dramatically described situations as better or worse than they actually were [11]. For sarcasm, annotators identified elements of contempt, often indicated by negative polarity words used to criticize someone [12]. Sarcasm, a form of irony, employs implied meanings to mock or deride [13]. We sought sentences with implied meanings that aimed to ridicule weaknesses or events negatively.

Irony was characterized by having two elements: a literal meaning and an implied meaning, with the two needing to differ to produce a humorous effect [14, 15].

Annotators followed the sequence defined in the codebook to ensure that categories such as sarcasm [16] and exaggeration [17], which are specific types of irony, were correctly classified only when their unique elements were present. Irony served as an overarching category encompassing these specific humor types, providing a structured framework for accurate annotation.

2.2. Multi-Class Classification with DeBERTa

In our study, we employ the DeBERTa [18] model as the base model for our experiments. DeBERTa has been recognized as one of the leading choices for encoder models due to its superior performance in various natural language processing tasks. We fine-tune the DeBERTa model using our training dataset and conduct two separate experimental runs. The first run involves using the dataset in its original form, without any modifications to address class imbalances. In this approach, we aim to evaluate the model's performance on the raw, imbalanced data. In the second experimental run, we address the class imbalance by implementing an under-sampling strategy. This method ensures that the representation of each class is balanced, preventing any single class from having a disproportionately high number of samples. Specifically, we cap the number of samples for the majority classes at $n = 250$. For the fine-tuning process, we utilize the *deberta-v3-large* model. The fine-tuning parameters are meticulously chosen to optimize performance. The learning rate is set to 2×10^{-5}, with a training batch size of 8 and an evaluation batch size of 16. The model is trained for 5 epochs, and we apply a weight decay of 0.01 to regularize the training process.

Table 1
Results on the Training Set (Weighted Average)

	Accuracy	Precision	Recall	F-Score	Support
Guided Annotation	0.7126	0.7383	0.7126	0.7148	87
DeBERTa	0.7983	0.8015	0.7983	0.7939	1715
DeBERTa$_{sampled}$	0.7854	0.8124	0.7854	0.7906	1715
GPT-4o	0.4496	0.5271	0.4496	0.4563	1715

Table 2
Results on the Test Set (Weighted Average)

	Accuracy	Precision	Recall	F-Score	Support
Guided Annotation	0.6667	0.7282	0.6667	0.6685	45
DeBERTa	0.6870	0.6844	0.6870	0.6731	722
DeBERTa$_{sampled}$	0.6787	0.6936	0.6787	0.6768	722
GPT-4o	0.4668	0.5275	0.4668	0.4733	722

2.3. Prompting with LLMs

In our methodology involving large language models (LLMs), we utilized GPT-4o [19], the most recent model developed by OpenAI. Our approach incorporated the few-shot prompting technique, which involves providing the model with a limited number of examples to guide its responses. Specifically, we included one example for each class, which served as a template to demonstrate the desired output format and content. Detailed descriptions of these prompts can be found in the appendix. The methodology was inspired from [20] where humor theories were embedded into the prompts. For reproducibility of our results, we set the random seed to 2024 and configured the temperature parameter to 0 as outlined in the OpenAI documentation. By setting the temperature to 0, we aimed to reduce the model's output variability, thereby enhancing consistency and repeatability in the generated responses.

3. Results

The metrics of precision, recall, accuracy, and F-score are reported and metrics in the tables below are computed for both the training and test dataset.

Table 1 and Table 2 present the weighted average performance metrics for the four approaches on both the training and test sets. On the training set, Guided Annotation shows moderate performance with an accuracy of 0.7126 and a balanced F-score of 0.7148, evaluated on a smaller subset (support of 87). DeBERTa exhibits the highest performance across all metrics, with an accuracy of 0.7983 and an F-score of 0.7939, indicating strong overall performance. DeBERTa$_{sampled}$ has slightly lower accuracy (0.7854) and F-score (0.7906) than DeBERTa but maintains high precision (0.8124). GPT-4o lags significantly with the lowest accuracy (0.4496) and F-score (0.4563), suggesting a need for change with the methodolgy with LLMs. On the test set, DeBERTa's performance decreases compared to the training set with an accuracy of 0.6870 and an F-score of 0.6731, indicating some loss in generalization. DeBERTa$_{sampled}$ also shows a decrease in performance with an accuracy of 0.6787 and an F-score of 0.6768, but it maintains higher precision than recall, suggesting it still identifies relevant instances well. GPT-4o again shows the lowest performance with an accuracy of 0.4668 and an F-score of 0.4733, consistent with its training performance.

Table 3 and Table 4 provide class-wise performance metrics for the approaches on both the training and test sets. Guided Annotation shows varying performance across classes, with high F-scores in SC (0.8718 on training, 0.9000 on test) and AID (0.8000 on training, 0.8333 on test), but very low performance in EX (0.1538 on training, 0.2222 on test) and WS (0.5000 on training, 0.4000 on test). DeBERTa performs consistently well across most classes, especially in AID (0.9671 on training, 0.8889 on test) and SD

Table 3
Results on the Training Set (By Individual Classes)

	Class	Precision	Recall	F-Score	Support
Guided Annotation	SD	0.6667	0.6667	0.6667	12
	WS	0.3333	1.0000	0.5000	3
	EX	0.2000	0.1250	0.1538	8
	IR	0.6842	0.7647	0.7222	17
	SC	1.0000	0.7727	0.8718	22
	AID	0.8000	0.8000	0.8000	25
DeBERTa	SD	0.8600	0.9430	0.8996	228
	WS	0.4797	0.5680	0.5201	125
	EX	0.5074	0.3286	0.3988	210
	IR	0.8226	0.7556	0.7877	356
	SC	0.5892	0.8765	0.7047	162
	AID	0.9837	0.9511	0.9671	634
DeBERTa$_{sampled}$	SD	0.7700	0.9693	0.8583	228
	WS	0.5305	0.6960	0.6021	125
	EX	0.5257	0.6333	0.5745	210
	IR	0.8393	0.6601	0.7390	356
	SC	0.7487	0.9012	0.8179	162
	AID	0.9795	0.8281	0.8974	634
GPT4o	SD	0.1905	0.2281	0.2076	228
	WS	0.2500	0.3120	0.2776	125
	EX	0.2530	0.4000	0.3100	210
	IR	0.7863	0.2893	0.4230	356
	SC	0.4803	0.8272	0.6077	162
	AID	0.6599	0.5662	0.6095	634

Table 4
Results on the Test Set (By Individual Classes)

	Class	Precision	Recall	F-Score	Support
Guided Annotation	SD	0.7500	0.6000	0.6667	5
	WS	0.5714	0.6667	0.6154	6
	EX	0.5000	0.1429	0.2222	7
	IR	0.2727	0.7500	0.4000	4
	SC	1.0000	0.8182	0.9000	11
	AID	0.8333	0.8333	0.8333	12
DeBERTa	SD	0.6777	0.9011	0.7736	91
	WS	0.4464	0.5102	0.4762	49
	EX	0.4255	0.1887	0.2614	106
	IR	0.5946	0.5986	0.5966	147
	SC	0.5000	0.8305	0.6242	59
	AID	0.9206	0.8593	0.8889	270
DeBERTa$_{sampled}$	SD	0.6833	0.9011	0.7773	91
	WS	0.4444	0.5714	0.5000	49
	EX	0.4144	0.4340	0.4240	106
	IR	0.6596	0.4218	0.5145	147
	SC	0.5185	0.7119	0.6000	59
	AID	0.9091	0.8519	0.8795	270
GPT4o	SD	0.2174	0.2747	0.2427	91
	WS	0.2642	0.2857	0.2745	49
	EX	0.2953	0.4151	0.3451	106
	IR	0.6716	0.3061	0.4206	147
	SC	0.4397	0.8644	0.5829	59
	AID	0.7117	0.5852	0.6423	270

(0.8996 on training, 0.7736 on test), but struggles in EX (0.3988 on training, 0.2614 on test) and WS (0.5201 on training, 0.4762 on test). DeBERTa$_{sampled}$ also shows strong performance, particularly in AID (0.8974 on training, 0.8795 on test) and SD (0.8583 on training, 0.7773 on test), with improved performance in EX (0.5745 on training, 0.4240 on test) and WS (0.6021 on training, 0.5000 on test) compared to its non-sampled counterpart, indicating better generalization.

4. Conclusions

The task of humor classification, particularly identifying specific types of humor, remains a complex challenge due to the subjective nature of humor perception. Our study presented a comprehensive approach involving guided annotation, fine-tuning the DeBERTa model, and using prompting with GPT-4o. The results highlight the effectiveness of DeBERTa in both original and sampled forms, showcasing its strong performance across various humor types. However, GPT-4o demonstrated significant limitations, suggesting that current LLMs may require further refinement or alternative methodologies to handle the nuances of humor classification effectively. Future research should focus on integrating the guided annotation approach directly into the prompting process for large language models (LLMs). By embedding detailed codebook guidelines and structural cues within the prompts, we can provide LLMs with more context and specificity, potentially improving their performance in humor classification tasks.

References

[1] W. Ruch, Psychology of humor, in: V. Raskin (Ed.), The Primer of Humor Research, number 8 in Humor Research, Mouton de Gruyter, Berlin, 2008, pp. 17–100. doi:10.1515/9783110198492.17.

[2] L. Ermakova, T. Miller, A.-G. Bosser, V. M. Palma Preciado, G. Sidorov, A. Jatowt, Overview of JOKER - CLEF-2024 track on Automatic Humor Analysis, in: L. Goeuriot, P. Mulhem, G. Quénot, D. Schwab, L. Soulier, G. M. Di Nunzio, P. Galuščáková, A. G. Seco de Herrera, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024), 2024.

[3] L. Ermakova, A.-G. Bosser, T. Miller, T. Thomas-Young, V. M. Palma Preciado, G. Sidorov, A. Jatowt, CLEF 2024 JOKER lab: Automatic Humour Analysis, in: N. Goharian, N. Tonellotto, Y. He, A. Lipani, G. McDonald, C. Macdonald, I. Ounis (Eds.), Advances in Information Retrieval: 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24–28, Proceedings, Part VI, volume 14613 of *Lecture Notes in Computer Science*, Springer, Cham, 2024, pp. 36–43. doi:10.1007/978-3-031-56072-9_5.

[4] V. M. Palma Preciado, et al., Overview of the clef 2024 joker task 2: Humour classification according to genre and technique, in: G. Faggioli, et al. (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2024), CEUR Workshop Proceedings, CEUR-WS.org, 2024. URL: http://ceur-ws.org.

[5] L. Ermakova, A.-G. Bosser, A. Jatowt, T. Miller, The joker corpus: English-french parallel data for multilingual wordplay recognition, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), Association for Computing Machinery, New York, NY, USA, 2023, pp. 2796–2806. doi:10.1145/3539618.3591885.

[6] D. L. Long, A. C. Graesser, Wit and humor in discourse processing, Discourse processes 11 (1988) 35–60.

[7] O. Couder, Problem solved? absurdist humour and incongruity-resolution, Journal of Literary Semantics 48 (2019) 1–21.

[8] A. Kamal, M. Abulaish, Self-deprecating humor detection: A machine learning approach, in: Computational Linguistics: 16th International Conference of the Pacific Association for Computational

Linguistics, PACLING 2019, Hanoi, Vietnam, October 11–13, 2019, Revised Selected Papers 16, Springer, 2020, pp. 483–494.

[9] R. A. Martin, P. Puhlik-Doris, G. Larsen, J. Gray, K. Weir, Individual differences in uses of humor and their relation to psychological well-being: Development of the humor styles questionnaire, Journal of research in personality 37 (2003) 48–75.

[10] E. Troiano, C. Strapparava, G. Özbal, S. S. Tekiroğlu, A computational exploration of exaggeration, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3296–3304.

[11] J. Patro, S. Baruah, A simple three-step approach for the automatic detection of exaggerated statements in health science news, in: P. Merlo, J. Tiedemann, R. Tsarfaty (Eds.), Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Association for Computational Linguistics, Online, 2021, pp. 3293–3305. URL: https://aclanthology.org/2021.eacl-main.289. doi:10.18653/v1/2021.eacl-main.289.

[12] S. K. Bharti, K. S. Babu, S. K. Jena, Parsing-based sarcasm sentiment recognition in twitter data, in: Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, 2015, pp. 1373–1380.

[13] J. D. Campbell, A. N. Katz, Are there necessary conditions for inducing a sense of sarcastic irony?, Discourse Processes 49 (2012) 459–480.

[14] H. P. Grice, Logic and conversation, in: Speech acts, Brill, 1975, pp. 41–58.

[15] D. Sperber, D. Wilson, Irony and the use-mention distinction, Philosophy 3 (1981) 143–184.

[16] H. L. Colston, Irony and sarcasm, in: The Routledge handbook of language and humor, Routledge, 2017, pp. 234–249.

[17] D. Wilson, D. Sperber, On verbal irony, Lingua 87 (1992) 53–76.

[18] P. He, J. Gao, W. Chen, DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing, in: The Eleventh International Conference on Learning Representations, 2023. URL: https://openreview.net/forum?id=sE7-XhLxHA.

[19] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., Gpt-4 technical report, arXiv preprint arXiv:2303.08774 (2023).

[20] R. R. Dsilva, Augmenting Large Language Models with Humor Theory To Understand Puns (2024). URL: https://hammer.purdue.edu/articles/thesis/Augmenting_Large_Language_Models_with_Humor_Theory_To_Understand_Puns/25674792. doi:10.25394/PGS.25674792.v1.

A. Prompts

```
### Instruction ###
You are an expert in linguistics and humor. Classify the text into one of the
    appropriate types of humor from the following: Irony (IR), Sarcasm (SC),
    Exaggeration (EX), Absurdity & Incongruity (AID), Self-Deprecating (SD), Wit (WS).
    You must respond with valid JSON with only one key ‘output‘, containing the correct
    classification of the sentence.

### Text ###
I tried to learn how to make puns, but no pun in ten did.
### Humor Type ###
{
  "output": "WS"
}

### Text ###
Did you hear about the pasta that got locked out of the house? Gnocci.
### Humor Type ###
{
  "output": "AID"
}

### Text ###
Amazing how fast this team can go winning from 13 straight to losing three in a row. Lol
    . Horrible managing tonight. I really hope this Boone experiment is over soon.
```

```
### Humor Type ###
{
  "output": "SC"
}

### Text ###
Good day, this is your trashcan speaking.
### Humor Type ###
{
  "output": "SD"
}

### Text ###
Wait so the founder of a thing says his thing is the best way to do a thing? Whoa.
### Humor Type ###
{
  "output": "IR"
}

### Text ###
Ohio news station reminds viewers what day it is during coronavirus lockdown.
### Humor Type ###
{
  "output": "EX"
}

### Text ###
<sentence>
### Humor Type ###
<response>
```

B. Guided Annotation Codebook

Please follow the following instructions to annotate the given sentences into the category most suitable according to the instructions given. Follow the order in which each category is described and move forward to the next category only if the previous category is eliminated.

Category 1

1. Identify words with multiple meanings or homophones (similar sounding words), supported by contextual clues within the sentence.
2. Note any unexpected elements or sudden changes in the sentence.

Category 2

1. Look for sentences structured as *?*. Or,
2. There exist words in the second part of the text which you might not expect on the basis of the first part of the sentence.
3. Identify phonetic incongruities in the second part of the text.
4. Detect illogical situations or events that are unrealistic or nonsensical.

Category 3

1. Look for events which might be embarrassing. Or,
2. Look for human flaws or weakness described and,
3. Look for specific sentence structures indicating self-reference, such as:
   a) Interjection followed by ’I’, ’we’, or ’you’.
   b) Conjunction followed by ’I’, ’we’, or ’you’.
   c) Question followed by ’I’ or ’we’.
   d) ’I’ or ’we’ followed by a verb.
   e) ’I’ or ’we’ followed by a negative model verb.
   f) Frequency of ’my’, ’me’, and ’I’.
   g) Presence of negative polarity.

Category 4

1. Identify situations or events described in a manner better or worse than normal. And,
2. Assess the description of events and their impacts for overly dramatic elements.

Category 5

1. Determine if the sentence conveys negative polarity, showing contempt or criticism. And,
2. Assess whether the sentence criticizes something or mocks a phenomenon or event. And,
3. Verify if the sentence's meaning differs from its literal interpretation.

Category 6

1. Identify the literal meaning of the sentence. And,
2. Discern any implied meanings, ensuring they differ from the literal interpretation.