PUNderstand: Using Large Language Models to Understand Puns

Ryan Rony Dsilva

dsilvar@purdue.edu

Abstract

This research investigates the computational processing of puns and humor within the framework of Large Language Models (LLMs). Grounded in the humor theory, the study explores the integration of humor principles into LLMs to enhance their proficiency in pun comprehension. The experiments employ datasets from SemEval-2017 Task 7, addressing pun detection and localization as classification tasks. Results reveal that the standard GPT-3.5 model outperforms smaller models, with the incorporation of humor prompts enhancing performance, particularly for smaller models. The study also delves into pun interpretation, highlighting challenges related to contextual understanding and potential model hallucinations. The findings underscore the need for continued research into prompt variations, data contamination, and the delicate balance between LLM size and computational demands. Our code is available on GitHub[1].

1 Overview and Motivation

According to T. Miller, Hempelmann, and Gurevych (2017): "A pun is a form of wordplay in which a word suggests two or more meanings by exploiting polysemy, homonymy, or phonological similarity to another word, for an intended humorous or rhetorical effect". English puns can be broadly classified into two categories: heterographic puns, which exploit the similarity in sound between words with different meanings, and homographic (homophonic) puns, which take advantage of the multiple meanings of a single word or phrase. An example of a heterographic pun is "I used to play piano by ear, but now I use my hands". The phrase "by ear" means playing music without sheet music or written instructions, but in this case, it is contrasted with "using my hands", which is a more literal way to play the piano. Furthermore, an example of a homographic pun is, "I'm reading a book on anti-gravity. It's impossible to put down". The phrase "put down" can mean to stop reading or to set down an object physically, but in the context of a book about anti-gravity, it also means to resist the book's upward pull.

Understanding of puns requires extra effort because the communicator uses ambiguous contextual assumption deliberately. Thus the audience cannot readily see the context (Gan, 2015). The primary motivation for studying puns in the context of Natural Language Processing (NLP) lies in the potential applications and advancements that computational humor can bring to various fields. Puns are commonly used in everyday conversations, comedy, literature, and advertising, among others (Gan, 2015). Understanding puns has a number of real-word applications like human computer interaction, sentiment analysis, machine assisted translation, and digital humanities (T. Miller & Turković, 2016).

2 Literature Review

2.1 Relevant Previous Work

2.1.1 Humor Theory

Humor theories (Attardo & Raskin, 1991; Raskin, 1979; Ritchie, 1999; Suls, 1972) concur that humor arises from incongruity present in a text. Incongruity theories posit that a

[1] https://github.com/RyanDsilva/punderstand

humorous text conveys two distinct and incompatible interpretations, often referred to as scripts or frames. These interpretations share a common element that facilitates the transition from one script to another. Typically, one interpretation is more apparent, leading the recipient to initially process only this script. However, as the text unfolds, certain elements contradict the initial interpretation, unveiling the previously concealed second interpretation (Amin & Burghardt, 2020).

Kao, Levy, and Goodman (2016) identify two factors contributing to humor in sentences: ambiguity and distinctiveness. While unambiguous sentences are less likely to be funny, mere ambiguity is insufficient for humor. The key lies in having distinct topical subsets or "viewpoints" within the sentence. For a sentence to be humorous, there must be sets of words supporting different interpretations, and these sets must be distinct from each other, creating a sense of incongruity.

He, Peng, and Liang (2019) introduce the "local-global surprisal principle" as a general principle for puns. The key observation is that the interpretation strength of a pun and alternative words reverses as one reads the sentence. For instance, in a given context, one word may be favored locally, while the alternative is favored globally. The proposed surprisal principle suggests that the pun word is significantly more surprising in the local context than in the global context, whereas the alternative word follows the opposite pattern.

2.1.2 Computational Systems for Puns

In the domain of pun detection, diverse methodologies have been employed. Pedersen (2017) utilized four configurations of a Word Sense Disambiguation (WSD) algorithm, identifying puns based on the presence of more than two distinct sense labels in the context. Xiu, Lan, and Wu (2017) adopted a supervised approach, training a classifier with features from WordNet (G. A. Miller, 1995) and word2vec embeddings. Pramanick and Das (2017) employed a hidden Markov model and cyclic dependency network with features from part-of-speech tagging and syntactic parsing. Multiple supervised systems integrated lexical-semantic and word embedding features for pun detection (Cattle & Ma, 2018; Diao et al., 2018; Jain, Yadav, & Javed, 2019). Dsilva (2023) implemented binary classification and sequence classification, respectively, using various features for pun detection. Neural network-based approaches were also explored, with Diao et al. (2020) emphasizing the enhancement of pun detection through contextualized and word representations.

Various methods have been employed for pun localization, encompassing diverse approaches. Oele and Evang (2017) approach involves sentence segmentation and utilizes a Lesk-like Word Sense Disambiguation (WSD) algorithm based on word and sense embeddings to identify homographic puns. Similarly, Huang, Huang, and Chen (2017) employs a WSD algorithm, relying on observations that a supporting word often realizes the second interpretation of a polysemous word and that puns are typically located toward the end of sentences. Statistical and machine learning-based techniques, such as those presented by Pramanick and Das (2017), utilize hidden Markov models and cyclic dependency networks, employing features from part-of-speech taggers and syntactic parsers. Knowledge and semantics-based approaches, such as Xiu et al. (2017)'s knowledge-based pun locator, leverage sense vectors and semantic similarity calculations. Rule-based methods, exemplified by Vechtomova (2017), employ heuristics considering word position, relation to punctuation, and corpus rarity. Neural network-based methods, including those proposed by Cai, Li, and Wan (2018) and Arroubat (2022), integrate word sense disambiguation and bidirectional LSTM networks or autoregressive models. Phonetic-based strategies, such as Doogan, Ghosh, Chen, and Veale (2017)'s use of Google n-grams and word2vec, and Liu et al. (2021)'s Dual-Attentive Neural Network, involve phonetic distance computation and contextualized representations for pun recognition and localization.

Several approaches have been proposed for pun interpretation. Oele and Evang (2017) focused on homographic puns, utilizing semantic similarity calculations among potential divisions of context and a Lesk-like Word Sense Disambiguation (WSD) algorithm. For homographic puns, alternative interpretations were obtained through various configurations of a WSD algorithm. Pedersen (2017) incorporated WordNet information, while Hurtado, Segarra, Pla, Carrasco, and González (2017) identified context words with embeddings similar to the pun and employed a bag-of-words representation for WSD. Jain et al. (2019) used word embeddings and the Lesk algorithm, and Liu et al. (2021) treated pun interpretation as a classification task. Several JOKER competition systems, such as those by Arroubat (2022); Brunelière, Germann, and Salina (2023); Ohnesorge, Gutiérrez, and Plichta (2023); Popova and Dadić (2023); Prnjak, Davari, and Schmitt (2023), applied large language models with zero-shot and very primitive few-shot prompting.

2.1.3 Large Language Models and Humor

Representing a noteworthy advancement in natural language processing and artificial intelligence, Large Language Models (LLMs) mark a significant milestone. Utilizing deep learning techniques, LLMs are pre-trained language models designed to process and comprehend natural language, showcasing remarkable capabilities in understanding and generating human-like text (Zhao et al., 2023).

Notably, emergent behaviors, such as the capacity for reasoning, become evident as these models reach a sufficient size (Qiao et al., 2022; Wei et al., 2022). LLMs are quite capable of making correct individual deduction steps, and so are generally capable of reasoning (Saparov & He, 2022). Using the right prompts is essential to obtain the favourable response from a language model (Jiang, Xu, Araki, & Neubig, 2020). Moreover, increasing the size of the model does not necessarily improve the results obtained as per the user's intent (Ouyang et al., 2022) and is dependant on many other factors such as quality of prompts, fine-tuning, etc.

The incorporation of humor into LLMs introduces a unique set of challenges and opportunities. LLMs have recently been employed for humor (Choi, Pei, Kumar, Shu, & Jurgens, 2023). While Large Language Models (LLMs) exhibit a formidable capability to generate linguistically complex and human-like text, their creativity and humor outputs have discernible limitations. To enhance their creative and humorous potential, leveraging explicit humor theories, by incorporating richer knowledge in prompts, may prove fruitful in overcoming these constraints Inácio and Oliveira (2023). In Jentzsch and Kersting (2023), three common characteristics—structure, wordplay, and topic—were identified. Merely having a single joke feature, like a question-answer template, does not lead to misclassification, indicating that ChatGPT possesses a discernment of humorous elements. However, the likelihood of a sample being classified as a joke increases with the presence of multiple joke characteristics, emphasizing the importance of a combination of these elements for accurate classification. Goes, Sawicki, Grześ, Brown, and Volpe (2023) also experimented using GPT-4 to evaluate jokes.

2.2 Contribution of Current Work

This research addresses the existing void in the integration of computational models for pun processing and humor. As highlighted by T. Miller et al. (2017), current computational systems for pun-related tasks lack a foundation in humor theories. The conjecture advanced in this study posits that incorporating principles from humor theories into these models will enhance their proficiency in comprehending puns. Additionally, recent studies (Ermakova et al., 2023, 2022) demonstrate that the prompts employed in large language models primarily consist of direct zero-shot instances. A secondary hypothesis posits that the formulation of well-crafted prompts, coupled with techniques such as few-shot prompting and fine-tuning, contributes to the enhanced performance of large language models.

3 Methodology

3.1 Datasets and Data Preparation

T. Miller and Gurevych (2015) produced a novel data set for SemEval-2017 Task 7 consisting of manually sense-annotated homographic puns with the following constraints: one pun per instance, one content word per pun, two meanings per pun and weak homography.

The dataset for SemEval 2017 Task 7 comprised two components. The homographic dataset encompassed 2250 contexts, with 1607 (71%) containing a pun. Among these 1607 puns, 1298 (81%) exhibited both meanings in WordNet. The second dataset mirrored the first, differing only in that the puns were heterographic instead of homographic. This dataset consisted of 1780 contexts, of which 1271 (71%) featured a pun. Among these puns, 1098 (86%) demonstrated dual meanings in WordNet.

3.2 Experiments

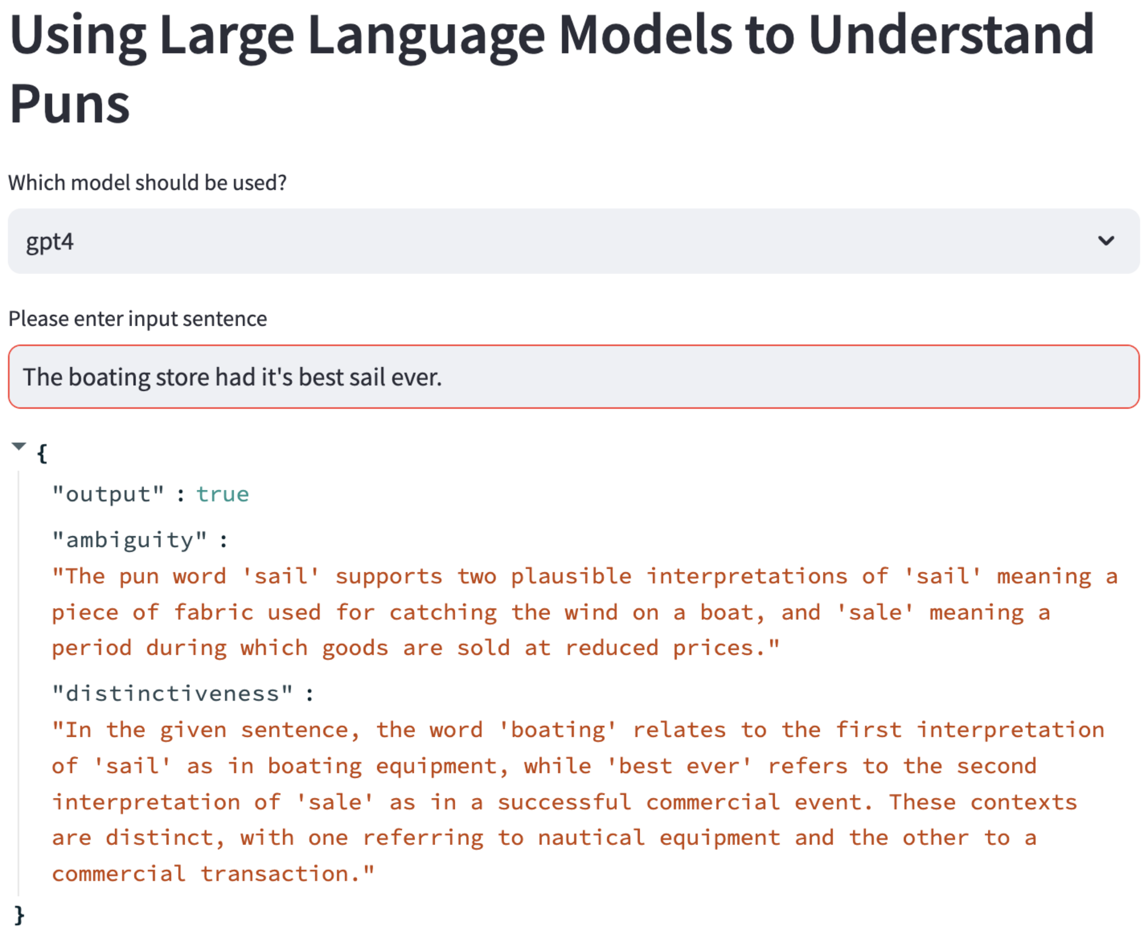
We embrace the computational humor theory proposed by Kao et al. (2016), wherein the assessment of humor is grounded in the quantitative evaluation of ambiguity and distinctiveness. Our approach involves the induction of a large language model (LLM) through the presentation of explicit definitions for both ambiguity and distinctiveness. It is asserted that the fulfillment of both conditions is not only necessary but also sufficient for the classification of a given sentence as a pun. The task is hence modelled as a binary classification task with the expected output to be a boolean.

- *Ambiguity* - there exists a word in the sentence that has two similarly likely interpretations.
- *Distinctiveness* - the two interpretations are very different from each other i.e. there are distinct words semantically related to the two interpretations. There needs to be at least one different word in the set of words supporting each interpretation.

In the context of pun location, the model is stimulated to generate the identified pun word, and subsequently, these outputs are juxtaposed with the gold labels that have been annotated within the dataset. The accuracy of the system's predictions is contingent upon the congruence between the predicted word or word location and the ground truth labels. In instances where the model's predictions align with the annotated labels, such occurrences are deemed correct assertions by the system.

The pursuit of pun interpretation is herein conceptualized as a task akin to text generation, thereby facilitating a qualitative analytical approach. It is noteworthy that methodologies for the seamless integration of WordNet, with Large Language Models (LLMs) are not well-established. Consequently, the evaluation of the interpretative quality necessitates a manual scrutiny, as there exists no predefined systematic framework for the amalgamation of WordNet with LLMs.

We use a reduced 25% subset of the SemEval 2017 Task 7 dataset in order to perform our experiments within the limited computational resources at our disposal.



Using Large Language Models to Understand Puns

Which model should be used?

gpt4

Please enter input sentence

The boating store had it's best sail ever.

```
{
  "output" : true
  "ambiguity" :
  "The pun word 'sail' supports two plausible interpretations of 'sail' meaning a
  piece of fabric used for catching the wind on a boat, and 'sale' meaning a
  period during which goods are sold at reduced prices."
  "distinctiveness" :
  "In the given sentence, the word 'boating' relates to the first interpretation
  of 'sail' as in boating equipment, while 'best ever' refers to the second
  interpretation of 'sale' as in a successful commercial event. These contexts
  are distinct, with one referring to nautical equipment and the other to a
  commercial transaction."
}
```

Figure 1: Screenshot of Demo Application

3.3 Technology Details

Our experimental procedures are conducted employing the Python programming language, with the Hugging Face transformers library[2] serving as the primary foundational framework. We also use Llama.cpp[3] for local inference on our systems. The instantiated models in our study, OpenOcra[4], are derived from the Mistral architecture, specifically the 7B variant. We also utilize the Open AI API for inference with GPT-3.5 and GPT 4. For the frontend application, we used the Streamlit[5] framework.

3.4 Demo Application

The demo application has been constructed utilizing the Streamlit framework and features a selection input for specifying the model to be employed, offering two alternatives: GPT-4 and OpenOcra. Subsequently, a text field is provided wherein users may input a pun sentence. This is succeeded by an output window that presents the JSON output generated by the selected model. For visual reference, a depiction of the demo application is presented in Figure 1.

[2] https://huggingface.co/transformers

[3] https://github.com/abetlen/llama-cpp-python

[4] https://huggingface.co/Open-Orca/Mistral-7B-OpenOrca

[5] https://streamlit.io/

	homographic				heterographic			
System	P	R	A	F_1	P	R	A	F_1
GPT-3.5	**0.8211**	**0.8259**	**0.8259**	**0.8223**	**0.8248**	**0.8292**	**0.8292**	**0.8261**
GPT-3.5$_{humor}$	0.8049	0.7940	0.7940	0.7979	0.7995	0.7775	0.7775	0.7842
OpenOcra-7B	0.7688	0.5577	0.5577	0.5639	0.7710	0.5753	0.5753	0.5845
OpenOcra-7B$_{humor}$	0.7847	0.7069	0.7069	0.7210	0.7382	0.6517	0.6685	0.6517

Table 1: Effect of humor theory prompt on pun detection

	homographic				heterographic			
System	C	P	R	F_1	C	P	R	F_1
GPT-3.5	1.0000	0.8234	0.8234	0.8234	1.0000	**0.8648**	**0.8648**	**0.8648**
GPT-3.5$_{humor}$	1.0000	**0.8507**	**0.8507**	**0.8507**	1.0000	0.8208	0.8208	0.8208
OpenOcra-7B	1.0000	0.6766	0.6766	0.6766	1.0000	0.6730	0.6730	0.6730
OpenOcra$_{humor}$	1.0000	0.6567	0.6567	0.6567	1.0000	0.6258	0.6258	0.6258

Table 2: Effect of humor theory prompt on pun location

4 Results

The evaluation for pun detection and location are straightforward as they are modelled as classification tasks. Hence, we adopt the same metrics of coverage, precision, recall, accuracy and F-score as per existing literature.

For pun interpretation, we refrain from a direct comparative assessment but instead prefer a qualitative analysis as an alternative means of evaluation. Few select examples are highlighted in 4.3.

4.1 Pun Detection

In our experimental investigations summarized in Table 1, the standard GPT-3.5 model demonstrated superior performance compared to smaller models and the GPT-3.5$_{humor}$. It is noteworthy that employing the humor prompt did not substantially diminish the performance of GPT-3.5, suggesting that the model's efficacy is primarily attributable to its inherent characteristics rather than the specific prompt utilized. Conversely, for a smaller model such as OpenOcra, the utilization of the humor prompt significantly enhanced performance in the pun detection task. This substantiates our contention that enhancing large language models with principles derived from humor theory holds merit. However, the impact of humor theory in conjunction with varying model sizes remains an under explored area of research that warrants further investigation.

4.2 Pun Location

In the context of pun localization, our empirical investigations yielded diverse outcomes. While GPT-3.5 emerged as the unequivocal victor, the impact of humor-inducing prompts did not attain statistical significance. Notably, humor prompts exhibited a marginal tendency to diminish the efficacy of the foundational models in the task of pun localization. The underlying cause of this observation remains uncertain, whether attributable to the nature of the prompts or the humor theory per se, is indeterminate based on the findings of our experiments. Our results are summarized in Table 2.

4.3 Pun Interpretation

In this initial illustration, each of the models successfully identified the pun word; however, the smaller OpenOcra models demonstrated a more accurate interpretation. The ambivalence inherent in the term “light” within the sentence encompasses both its luminescent connotation and its connotation denoting smallness. The terms “manufacturer” and “lamps” correspond to the former, while “employees” and “workload” align with the latter. In contrast, the GPT-3.5 models, irrespective of the presence of a humor prompt, either correctly identified only one set of sense words or inaccurately identified both sets.

```
"text":"A manufacturer that made lamps gave their employees a light workload."
"pun_word": "light"
"target_word": "light"
"source_sense": "manufacturer;lamps"
"target_sense": "employees;workload"
```

In the following illustration, each model failed to accurately identify the phonological attribute connecting “whiskey” with “risky”. However, it is noteworthy that all models successfully discerned the semantic nuances, reflecting a situation where the model exhibited proficiency in contextual interpretation despite its inability to predict the specific target word. This observation prompts a concern regarding the extent to which the model authentically comprehended the context, raising the question of whether its successful interpretations were a result of genuine understanding or mere conjecture.

```
"text": "Making your own hard liquor is a whiskey business."
"pun_word": "whiskey"
"target_word": "whiskey"
"source_sense": "business"
"target_sense": "hard;liquor"
```

5 Discussion and Conclusion

In this study, we have delved into Large Language Models (LLMs) and naturally encountered several critical considerations that warrant attention in their application. Foremost among these concerns is the potential manifestation of hallucinations, an issue that has garnered increased scrutiny as LLMs become more prevalent in various domains. Hallucinations, or the generation of inaccurate or fictitious information, pose a significant challenge to the reliability and trustworthiness of LLM outputs, demanding vigilant scrutiny and the implementation of robust validation mechanisms.

Another pivotal concern that emerged from our investigation pertains to data contamination, as the provenance of the training data remains elusive. The opacity surrounding whether an LLM has encountered specific data during its training raises questions about the integrity and independence of its responses. This lack of transparency poses inherent challenges in ensuring data purity, thereby necessitating the establishment of protocols to ascertain and manage potential data contamination in LLM-generated content.

We understand the impact of prompt variations on LLM responses, acknowledging the intricate relationship between the input stimuli and generated output. Despite our adherence to best practices in prompt design, the inherent variability in responses introduces an element of uncertainty into the interpretation of results. The multifaceted nature of language prompts underscores the need for continued research into the nuanced interplay between prompt variations and LLM performance to enhance the reproducibility and reliability of findings. Furthermore, fine-tuning base large language models specifically for this domain may greatly improve performance as previously demonstrated in various works.

While this study did not explicitly focus on model size, the literature suggests that larger models may possess an enhanced capacity for nuanced reasoning tasks. However, the scalability of LLMs introduces additional computational demands, prompting a delicate balance between model size, performance, and resource constraints.

References

Amin, M., & Burghardt, M. (2020). A survey on approaches to computational humor generation. In *Proceedings of the the 4th joint sighum workshop on computational linguistics for cultural heritage, social sciences, humanities and literature* (pp. 29–41).

Arroubat, H. (2022). Wordplay location and interpretation with deep learning methods. In *Clef (working notes)* (pp. 1701–1705).

Attardo, S., & Raskin, V. (1991). Script theory revis (it) ed: Joke similarity and joke representation model.

Brunelière, O., Germann, C., & Salina, K. (2023). Clef 2023 joker task 2: using chat gpt for pun location and interpretation. *Proceedings of the Working Notes of CLEF*.

Cai, Y., Li, Y., & Wan, X. (2018). Sense-aware neural models for pun location in texts. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 546–551).

Cattle, A., & Ma, X. (2018). Recognizing humour using word associations and humour anchor extraction. In *Proceedings of the 27th international conference on computational linguistics* (pp. 1849–1858).

Choi, M., Pei, J., Kumar, S., Shu, C., & Jurgens, D. (2023). Do llms understand social knowledge? evaluating the sociability of large language models with socket benchmark. *arXiv preprint arXiv:2305.14938*.

Diao, Y., Lin, H., Yang, L., Fan, X., Wu, D., & Xu, K. (2020). Crga: Homographic pun detection with a contextualized-representation: Gated attention network. *Knowledge-Based Systems*, *195*, 105056.

Diao, Y., Yang, L., Zhang, D., Xu, L., Fan, X., Wu, D., & Lin, H. (2018). Homographic puns recognition based on latent semantic structures. In *Natural language processing and chinese computing: 6th ccf international conference, nlpcc 2017, dalian, china, november 8–12, 2017, proceedings 6* (pp. 565–576).

Doogan, S., Ghosh, A., Chen, H., & Veale, T. (2017). Idiom savant at semeval-2017 task 7: Detection and interpretation of english puns. In *Proceedings of the 11th international workshop on semantic evaluation (semeval-2017)* (pp. 103–108).

Dsilva, R. R. (2023). Akranlu@ clef joker 2023: using sentence embeddings and multilingual models to detect and interpret wordplay. *Proceedings of the Working Notes of CLEF*.

Ermakova, L., Miller, T., Bosser, A.-G., Palma Preciado, V. M., Sidorov, G., & Jatowt, A. (2023). Overview of joker–clef-2023 track on automatic wordplay analysis. In *International conference of the cross-language evaluation forum for european languages* (pp. 397–415).

Ermakova, L., Miller, T., Regattin, F., Bosser, A.-G., Borg, C., Mathurin, É., … others (2022). Overview of joker@ clef 2022: Automatic wordplay and humour translation workshop. In *International conference of the cross-language evaluation forum for european languages* (pp. 447–469).

Gan, X. (2015). A study of the humor aspect of english puns: views from the relevance theory. *Theory and Practice in Language Studies*, *5*(6), 1211.

Goes, F., Sawicki, P., Grześ, M., Brown, D., & Volpe, M. (2023). Is gpt-4 good enough to evaluate jokes? In *Proceedings of the 14th international conference for computational* *creativity*.

He, H., Peng, N., & Liang, P. (2019). Pun generation with surprise. In *2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, naacl hlt 2019* (pp. 1734–1744).

Huang, Y.-H., Huang, H.-H., & Chen, H.-H. (2017). Identification of homographic pun location for pun understanding. In *Proceedings of the 26th international conference on world wide web companion* (pp. 797–798).

Hurtado, L.-F., Segarra, E., Pla, F., Carrasco, P., & González, J.-A. (2017). Elirf-upv at semeval-2017 task 7: Pun detection and interpretation. In *Proceedings of the 11th international workshop on semantic evaluation (semeval-2017)* (pp. 440–443).

Inácio, M. L., & Oliveira, H. G. (2023). Towards generation and recognition of humorous texts in portuguese. In *Proceedings of the 17th conference of the european chapter of the association for computational linguistics: Student research workshop* (pp. 26–36).

Jain, A., Yadav, P., & Javed, H. (2019). Equivoque: detection and interpretation of english puns. In *2019 8th international conference system modeling and advancement in research trends (smart)* (pp. 262–265).

Jentzsch, S., & Kersting, K. (2023). Chatgpt is fun, but it is not funny! humor is still challenging large language models. *arXiv preprint arXiv:2306.04563*.

Jiang, Z., Xu, F. F., Araki, J., & Neubig, G. (2020). How can we know what language models know? *Transactions of the Association for Computational Linguistics*, *8*, 423–438.

Kao, J. T., Levy, R., & Goodman, N. D. (2016). A computational model of linguistic humor in puns. *Cognitive science*, *40*(5), 1270–1285.

Liu, S., Ma, M., Yuan, H., Zhu, J., Wu, Y., & Lan, M. (2021). A dual-attention neural network for pun location and using pun-gloss pairs for interpretation. In *Natural language processing and chinese computing: 10th ccf international conference, nlpcc 2021, qingdao, china, october 13–17, 2021, proceedings, part i 10* (pp. 688–699).

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, *38*(11), 39–41.

Miller, T., & Gurevych, I. (2015). Automatic disambiguation of english puns. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)* (pp. 719–729).

Miller, T., Hempelmann, C. F., & Gurevych, I. (2017). Semeval-2017 task 7: Detection and interpretation of english puns. In *Proceedings of the 11th international workshop on semantic evaluation (semeval-2017)* (pp. 58–68).

Miller, T., & Turković, M. (2016). Towards the automatic detection and identification of english puns. *The European Journal of Humour Research*, *4*(1), 59–75.

Oele, D., & Evang, K. (2017). Buzzsaw at semeval-2017 task 7: Global vs. local context for interpreting and locating homographic english puns with sense embeddings. In *Proceedings of the 11th international workshop on semantic evaluation (semeval-2017)* (pp. 444–448).

Ohnesorge, F., Gutiérrez, M., & Plichta, J. (2023). Clef 2023 joker tasks 2 and 3: using nlp models for pun location, interpretation and translation. *Proceedings of the Working Notes of CLEF*.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., … others (2022). Training language models to follow instructions with human feedback, 2022. *URL https://arxiv. org/abs/2203.02155*, *13*.

Pedersen, T. (2017). Duluth at semeval-2017 task 7: Puns upon a midnight dreary, lexical semantics for the weak and weary. In *Proceedings of the 11th international workshop on semantic evaluation (semeval-2017)* (pp. 416–420).

Popova, O., & Dadić, P. (2023). Does ai have a sense of humor? clef 2023 joker tasks 1, 2 and 3: using bloom, gpt, simplet5, and more for pun detection, location, interpretation and translation. *Proceedings of the Working Notes of CLEF*.

Pramanick, A., & Das, D. (2017). Ju cse nlp@ semeval 2017 task 7: Employing rules to detect and interpret english puns. In *Proceedings of the 11th international workshop on semantic evaluation (semeval-2017)* (pp. 432–435).

Prnjak, A., Davari, D. R., & Schmitt, K. (2023). Clef 2023 joker task 1, 2, 3: pun detection, pun interpretation, and pun translation. *Proceedings of the Working Notes of CLEF*.

Qiao, S., Ou, Y., Zhang, N., Chen, X., Yao, Y., Deng, S., ... Chen, H. (2022). Reasoning with language model prompting: A survey. *arXiv preprint arXiv:2212.09597*.

Raskin, V. (1979). Semantic mechanisms of humor. In *Annual meeting of the berkeley linguistics society* (Vol. 5, pp. 325–335).

Ritchie, G. (1999). *Developing the incongruity-resolution theory* (Tech. Rep.).

Saparov, A., & He, H. (2022). Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *arXiv preprint arXiv:2210.01240*.

Suls, J. M. (1972). A two-stage model for the appreciation of jokes and cartoons: An information-processing analysis. *The psychology of humor: Theoretical perspectives and empirical issues*, *1*, 81–100.

Vechtomova, O. (2017). Uwaterloo at semeval-2017 task 7: Locating the pun using syntactic characteristics and corpus-based metrics. In *Proceedings of the 11th international workshop on semantic evaluation (semeval-2017)* (pp. 421–425).

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., ... others (2022). Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Xiu, Y., Lan, M., & Wu, Y. (2017). Ecnu at semeval-2017 task 7: Using supervised and unsupervised methods to detect and locate english puns. In *Proceedings of the 11th international workshop on semantic evaluation (semeval-2017)* (pp. 453–456).

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., ... others (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*.

A Prompts

A.1 Pun Detection - No Humor Theory

The prompt describes the task and then includes some few shot examples, one for each case.

```
### INSTRUCTION:
You are to classify whether a given sentence is a pun:
Here are examples of the cases you will encounter:
1. An example of a pun -
The magician got so mad that he pulled his hare out.
{{
  "output": true,
  "explanation": "The pun word 'hare' supports two plausible interpretations of '
      hare' meaning a rabbit and 'hair' meaning human hair. In the given sentence
      , the words 'magician' relates to 'hare' while 'pulled' refers to the
      second interpretation of 'hair'. Both of these are distinct where one
      refers to a magician's animal while the other refers to an action done in
      anger which is pulling your hair."
}}
2. An example of no pun -
I went to the bank.
{{
  "output": false,
  "explanation": "The word 'bank' does have ambiguity here where it supports two
       plausible interpretations of bank as in a financial institution and bank
      as in the banks of a river. There are no other words in the sentence that
      realize both interpretations, hence the sentence is not a pun."
}}
3. An example of no pun -
Let us go home.
{{
  "output": false,
  "explanation": "There is no pun word in the sentence."
}}
Identify whether the input sentence is a pun and explain the result in valid
    JSON format. Generate a response in the form of a valid JSON object with two
     keys: output and explanation.

### INPUT: {{sentence}}

### OUTPUT:
```

A.2 Pun Detection - Humor Theory

The prompt first defines the two terms of Ambiguity and Distinctiveness followed by stating that both conditions are necessary and sufficient for a sentence to be classified as a pun. The prompt then includes some few shot examples, one for each case.

```
### INSTRUCTION:
You are to classify whether a given sentence is a pun given the criteria below:
1. Ambiguity - there exists a word in the sentence that has two similarly likely
     interpretations.
2. Distinctiveness - the two interpretations are very different from each other
    i.e. how distinct are the words semantically related to the two
    interpretations from each other. There needs to be at least one different
    word in the set of words supporting each interpretation.

For a given sentence to be a pun, it should satisfy BOTH criteria - Ambiguity
    and Distinctiveness. It is possible that a sentence does not have a word
    that is ambiguous. The result in this case is false. It is possible that a
    sentence has Ambiguity but no Distinctiveness. The result in this case is
    false.

Here are three examples of the cases you will encounter.
1. An example with both ambiguity and distinctiveness:
The magician got so mad that he pulled his hare out.
{{
  "output": true,
  "pun_word": "hare",
  "ambiguity": "The pun word ’hare’ supports two plausible interpretations of ’
      hare’ meaning a rabbit and ’hair’ meaning human hair.",
  "distinctiveness": "In the given sentence, the words ’magician’ relates to ’
      hare’ while ’pulled’ refers to the second interpretation of ’hair’. Both of
       these are distinct where one refers to a magician’s animal while the other
       refers to an action done in anger which is pulling your hair."
}}

2. An example with only ambiguity:
I went to the bank.
{{
  "output": false,
  "pun_word": "",
  "ambiguity": "The word ’bank’ does have ambiguity here where it supports two
      plausible interpretations of bank as in a financial institution and bank as
       in the banks of a river.",
  "distinctiveness": "There are no other words in the sentence that provide
      distinctiveness to the two interpretations, the sentence is not a pun."
}}

3. An example with neither ambiguity nor distinctiveness:
Let us go home.
{{
  "output": false,
  "pun_word": "",
  "ambiguity": "There is no ambiguious word in the sentence.",
  "distinctiveness": "Not Applicable"
}}

Identify whether the given sentence is a pun and explain the result based on
    ambiguity and distinctiveness in valid JSON format.

### INPUT: {{sentence}}

### OUTPUT:
```

A.3 Pun Location and Interpretation - No Humor Theory

The prompt describes the task and then includes some few shot examples, one for each case.

```
### INSTRUCTION:
You are to identify the pun word in a given sentence. It is guaranteed that the
    sentence has a pun word.
Also, list all the words that realize each interpretation of the pun word. If
    there are multiple words that support an interpretation, separate them with
    a semicolon.

Here are two examples of the cases you will encounter:
1. An example with heterographic puns (words that sound similar) -
The magician got so mad that he pulled his hare out.
```

```
{{
 "pun_word": "hare",
 "target_word": "hair",
 "source_sense": "magician",
 "target_sense": "angry;pulled"
}}
2. An example with homographic puns (same words with different senses) -
I used to be a banker but I lost interest.
{{
 "pun_word": "interest",
 "target_word": "interest",
 "source_sense": "banker",
 "target_sense": "used;to;be"
}}
Identify the pun word in the input sentence and predict the target word. Also,
    pick the supporting words for each sense. Generate a response in the form of
     a valid JSON object with four keys: pun_word, target_word, source_sense and
     target_sense.

### INPUT: {{sentence}}

### OUTPUT:
```

A.4 Pun Location and Interpretation - Humor Theory

The prompt first defines the two terms of Ambiguity and Distinctiveness followed by stating that both conditions are necessary and sufficient for a sentence to be classified as a pun. The prompt then includes some few shot examples, one for each case.

```
### INSTRUCTION:
You are to identify the pun word in a given sentence based on the following
    criteria:
1. Ambiguity - there exists a word in the sentence that has two similarly likely
     interpretations.
2. Distinctiveness - the two interpretations are very different from each other
    i.e. how distinct are the words semantically related to the two
    interpretations from each other. There needs to be at least one different
    word in the set of words supporting each interpretation.
For a word to be a pun word, it should satisfy BOTH criteria - Ambiguity and
    Distinctiveness.
It is guaranteed that the sentence has a pun word and hence the selected pun
    word must be from the sentence in the same form as it appears.
Also, list all the words that realize each interpretation of the pun word. If
    there are multiple words that support an interpretation, separate them with
    a semicolon.

Here are two examples of the cases you will encounter:
1. An example with heterographic puns (words that sound similar) -
The magician got so mad that he pulled his hare out.
{{
 "pun_word": "hare",
 "target_word": "hair",
 "source_sense": "magician",
 "target_sense": "angry;pulled"
}}

In the above sentence, the pun word 'hare' supports two plausible
    interpretations of 'hare' meaning a rabbit and 'hair' meaning human hair. In
     the given sentence, the words 'magician' relates to 'hare' while 'angry'
    and 'pulled' refer to the second interpretation of 'hair'.
```

```
2. An example with homographic puns (same words with different senses) -
I used to be a banker but I lost interest.
{{
 "pun_word": "interest",
 "target_word": "interest",
 "source_sense": "banker",
 "target_sense": "used;to;be"
}}

In the above sentence, the pun word 'interest' supports two distinct
    interpretations of interest meaning 'financial interest' and interest
    meaning 'desire to do something'. The words 'banker' realize the first sense
     of financial interest while the words 'used', 'to', 'be' refer to the
    second sense.

Identify the pun word in the input sentence is a pun and list the target word
    and the supporting words for each sense the result based on ambiguity and
    distinctiveness in valid JSON format. Do not explain. Generate a response in
     the form of a valid JSON object with four keys: pun_word, target_word,
    source_sense and target_sense.

### INPUT: {{sentence}}

### OUTPUT:
```