# Can GPT-4 Detect Euphemisms across Multiple Languages?

**Todd Firsich** and **Anthony Rios**
Department of Information Systems and Cyber Security
The University of Texas at San Antonio
todd.firsich@utsa.edu, anthony.rios@utsa.edu

## Abstract

Euphemisms are words or phrases used instead of another word or phrase that might be considered harsh, blunt, unpleasant, or offensive. Euphemisms generally soften the impact of what is being said, making it more palatable or appropriate for the context or audience. Euphemisms can vary significantly between languages, reflecting cultural sensitivities and taboos, and what might be a mild expression in one language could carry a stronger connotation in another. This paper uses prompting techniques to evaluate GPT-4 for detecting euphemisms across multiple languages as part of the 2024 FigLang shared task. We evaluate both zero-shot and few-shot approaches. Our method achieved an average macro F1 of .732, ranking first in the competition. Moreover, we found that GPT-4 does not perform uniformly across all languages, with a difference of .233 between the best (English .831) and the worst (Spanish .598) languages.

## 1 Introduction

A euphemism is a term or expression substituted for another that may be deemed too direct, harsh, or offensive. Euphemisms play a nuanced role in linguistic expression, serving as a polite or softer alternative to potentially sensitive or direct language (Danescu-Niculescu-Mizil et al.; Magu and Luo). However, their inherent ambiguity challenges Natural Language Processing (NLP) systems in comprehending meaning because they must pick up on subtle contextual cues (Bisk et al.; Carbonell and Minton). This difficulty is magnified in multilingual contexts, where the same euphemism could have different meanings across cultures. Hence, this paper describes an approach for the 2024 FigLan shared task for multilingual euphemism detection.

Much of the recent research on euphemism detection has focused on fine-tuning transformer-based models (Zhu and Bhat, 2021; Maimaitituoheti et al., 2022; Wang et al., 2022). For instance, Wang et al. (2022) combined a BERT-based transformer with a relational graph attention network and fine-tuned it for euphemism detection. However, recent advancements in the development of large language models (LLMs) like GPT-4 have been shown to be successful in similar tasks such as offensive and abusive language detection (OpenAI et al.; Wu et al.; Matter et al., 2024; Li et al., 2023). GPT-4 is supposedly trained on extensive datasets of multilingual text containing wide variations of linguistic styles, which would be very helpful in understanding and interpreting euphemistic language. The tool's ability to generate human-like dialogue and adapt itself to nuanced language suggests that it could be used to distinguish between literal and euphemistic language use.

Recent research has shown limitations of GPT-4 and related models in multi-lingual settings (Zhang et al., 2024; Ahuja et al., 2023). For example, Qiu et al. (2024) report substantial differences in medical applications performance of GPT-4 across different languages. Hence, understanding how GPT-4 performs for multilingual classification, particularly for tasks that involve figurative language, can provide unique insights into its limitations.

In this paper, we explore the application of prompting techniques (Ouyang et al.; Lester et al.; Liu et al.) to detect euphemisms using GPT-4. We note that recent work has explored prompting-based euphemism detection (Maimaitituoheti et al., 2022). However, the system still required fine-tuning model parameters. Here, we explore zero-shot and few-shot prompting strategies without any fine-tuning. We analyze a various number of in-context examples. Moreover, we performed a small error analysis to understand the limitations of GPT-4 for euphemism detection and to understand when GPT-4 fails for multilingual euphemism detection.



*Proceedings of the 4th Workshop on Figurative Language Processing (FLP)*, pages 65–72

## 2 Related Work

Despite the general advancements in NLP, the automated detection of euphemisms remains a relatively under-explored area. Early approaches to identify euphemistic speech focused on rule-based systems and statistical methods (Felt and Riloff). Keh et al. (2022) explored kNN and data augmentation for euphemism detection. Likewise, fine-tuning pretrained transformer models is a popular approach. For instance, Wiriyathammabhum (2022) fine-tune RoBERTa (Liu et al., 2019) models for euphemisim detection. Trust et al. (2022) combined RoBERTa models with cost-sensitive learning to handle class imbalance issues. Wang et al. (2022) combined a BERT-based transformer with a relational graph attention network and fine-tuned it for euphemism detection. However, these approaches cannot capture euphemisms' nuanced nature or how euphemisms change over time. With the advent of models such as BERT and its successors, researchers have been able to show the potential for neural network models to understand complex language phenomena like metaphors, sarcasm, and idioms (Magu and Luo; Wang et al.; Zhu and Bhat; Gavidia et al.).

While the LLMs have shown to be more capable, researchers identified that not only the size of the model and the training data used are important, but how a task is presented to the LLM is equally important (Wei et al.; Li et al.). Prompting offers a few benefits over fine-tuning a LLM. Prompting does not require a model to undergo an additional round of training, making it more resource-efficient and accessible. Also, prompting leverages the model's pre-trained knowledge, enabling quick adaptation to new tasks without the risk of overfitting. Prompting is particularly appealing for subtle language tasks like euphemism disambiguation, allowing the LLM to focus on the subtleties of euphemistic language without extensive training.

A few researchers have used prompting in previous euphemism studies (Keh; Maimaitituoheti et al.). Maimaitituoheti et al. used a RoBERTa model and fine-tuned the model to improve its performance using prompts. The most similar work to this paper is by Keh (2022), which used an older GPT-3 model and post-processing rules to classify the evaluation as euphemistic or literal. Their work found that fine-tuned models (e.g., RoBERTa) outperformed zero-shot and few-shot methods using GPT-3. In this work, we extend the idea of using prompting in two ways. First, we use GPT-4, which is more capable than GPT3-3. Second, this model is evaluated on the new multilingual euphemism dataset.

## 3 Methodology

In this section, we discuss the general task, dataset, and our prompting strategy. Overall, we use a few-shot prompting framework for our submission.

**Task.** The Multilingual Euphemism Detection Shared Task for the Fourth Workshop on Figurative Language Processing involves predicting whether a substring within a sentence is a euphemism. Specifically, given a string, "*This summer, the budding talent agent was <PET>between jobs</PET> and free to babysit pretty much any time*," participants need to detect whether the embedded Potential Euphemistic Terms (PET) is a euphemism or not for this specific context. This means that each PET can be a literal (not a euphemism or a euphemism). The participants' results are collected and evaluated on the shared task site at Codabench.[1]

**Dataset.** For this shared task, two sets of data are provided, each consisting of samples in Chinese, English, Spanish, and Yorùbá. The first sets are the training datasets to help refine the participants' methodology, consisting of rows of sentences, the embedded PET, and a classification label (euphemism or not). The composition of the datasets by language is provided in Table 1. The second set is the test dataset, which consists of only sentences and the embedded PET without ground truth labels. The composition of these datasets by language is also provided in Table 1. It was observed that the PETs in the training and test datasets match relatively often. For instance, we may find both "passed away" in the test and training data. Only 47 of the 67 PETs from the test dataset are in the training dataset for English. Each English PET in the test data matched an average of 1.83 euphemisms and 1.54 literal PETS. For Spanish, there are no PETs in the test dataset that are also in the training dataset. The Chinese dataset has 7 of the 48 PETs in both datasets (.38 euphemisms and .29 literal PETs on average), and Yorùbá has 14 of the 28 PETs in both datasets (0.41 euphemisms and .30 literal PETs on average). We split the training datasets into both a training and validation dataset, with 20% used for validation and 80% used as train-

[1] https://www.codabench.org/competitions/1959

| Language-Set | PETs | Num Sent. | Euph. |
|---|---|---|---|
| Chinese-Train | 111 | 2005 | 1484 |
| Chinese-Test | 48 | 1226 | — |
| English-Train | 163 | 1952 | 1383 |
| English-Test | 67 | 1196 | — |
| Spanish-Train | 147 | 1861 | 1143 |
| Spanish-Test | 85 | 1091 | — |
| Yorùbá-Train | 133 | 1941 | 1281 |
| Yorùbá-Test | 28 | 669 | — |

Table 1: Dataset Composition for Training and Testing

ing examples (i.e., to find matching PETs).

**Prompt Development.** We use a few-shot prompting framework for our approach. Specifically, we prompt GPT-4 using the OpenAI API to predict whether a given PET is either a euphemism (True), or not (False). We provide the prompt template below:

*Given the context, determine if the phrase 'PET' is used as a Euphemism. Reply with the word 'True' if it is used as a Euphemism in this context else 'False'.*

***«context»***

*A euphemism is a mild or indirect word or expression substituted for one considered to be too harsh, blunt, or offensive. Euphemisms are used to avoid directly mentioning unpleasant or taboo topics, and they are often employed to soften the impact of the information being conveyed*

***«Euphemism examples»***

*Example - Is the phrase '{PET}' a Euphemism in the following text. {text} — Answer - 'True'*

*Example - Is the phrase '{PET}' a Euphemism in the following text. {text} — Answer - 'True'*

***«Literal examples»***

*Example - Is the phrase '{PET}' a Euphemism in the following text. {text} — Answer - 'False'*

*Example - Is the phrase '{PET}' a Euphemism in the following text. {text} — Answer - 'False'*

***«task»***

*Given the context, is the phrase '{PET}' used as a Euphemism in the following text? Context: {Text}*

The prompt has five main components: instruction, context, examples of euphemism, and literal examples. The instruction provides the high-level task (e.g., return True or False). The context defines euphemisms. The euphemism and literal examples are instances directly from the training dataset. Each example is formatted in the form of "*Is the phrase [PET] a Euphemism in the following text [text].*" The PET is the substring of interest, e.g., 'between jobs." The text is the actual context that the PET appears in, e.g., "*the budding talent agent was <PET>between jobs</PET> and free to babysit pretty much any time.*" Each example is followed by a "*Label*" token and either a "*True*" or "*False*" value. Finally, the task is a single test instance that we wish to classify as either the PET being a euphemism or not.

For the study, five different styles of prompting were examined. The first style is "Zero-Shot," which only uses the instruction and the task. "Zero-Shot with context" adds the context information. Next is the "Few-Shot with Random Examples" method, which uses only one random euphemism and one literal example. Research suggests that better prompt performance is achieved when similar examples are provided to the LLM in the prompt (Wei et al.; Brown et al.). Hence, we also experiment with variations called "Few-Shot with Targeted Examples," where we use $k$ euphemism and $k$ literal examples with the same PET as the text instance. Specifically, if the text instance's PET is "between jobs," then we will find both up to $k$ euphemism and $k$ literal examples that also have the "between jobs" PET. If there are no other matching examples with the same PET, or there are fewer than $k$ matching examples, we choose the remaining examples at random.

**Experimental Details.** The process to evaluate the PETs used the GPT-4 APIs provided by OpenAI (OpenAI, 2023). The GPT-4 model used in our experiments is the "gpt-4-0125-preview" version and the processing occurred between 2024-02-06 and 2024-03-07. The model temperature was set at "0" to make the model less random. All other model parameters were accepted at their default values. The software developed to process each sample using the APIs was written in Python based on examples provided on the OpenAI developer website.[2]

## 4 Results

In this section, we report the results on both the validation and test datasets.

**Validation Dataset Results.** The validation dataset results are shown in Table 2. In total, we executed

[2]https://platform.openai.com/docs/guides/text-generation

| Technique | Language | F1 | Precision | Recall |
|---|---|---|---|---|
| Zero-Shot | Chinese | .650 | .581 | **.962** |
| Zero-Shot w context | Chinese | .748 | .916 | .795 |
| Few Shot - Ran. Examples | Chinese | .760 | .906 | .832 |
| Few Shot - Targ. Examples (2) | Chinese | .801 | .941 | .838 |
| Few Shot - Targ Examples (8) | Chinese | **.858** | **.957** | .891 |
| Zero-Shot | English | .707 | **.912** | .675 |
| Zero-Shot \w context | English | .732 | .861 | .805 |
| Few Shot - Ran. Examples | English | .715 | .841 | .819 |
| Few Shot - Targ. Examples (2) | English | .747 | .877 | .801 |
| Few Shot - Targ. Examples (8) | English | **.820** | .907 | **.877** |
| Zero-Shot | Spanish | .545 | .794 | .345 |
| Zero-Shot + context | Spanish | .666 | .800 | .592 |
| Few Shot - Ran. Examples | Spanish | .662 | .772 | .623 |
| Few Shot - Targ. Examples (2) | Spanish | .698 | .825 | .632 |
| Few Shot - Targ. Examples (8) | Spanish | **.761** | **.911** | **.776** |
| Zero-Shot | Yorùbá | .400 | 1.000 | .181 |
| Zero-Shot with context | Yorùbá | .610 | .926 | .498 |
| Few Shot - Ran. Examples | Yorùbá | .674 | .923 | .61 |
| Few Shot - Targ. Examples (2) | Yorùbá | .761 | .911 | .776 |
| Few Shot - Targ. Examples (8) | Yorùbá | **.872** | **.951** | **.916** |

Table 2: F1, Precision, and Recall for each prompting technique for each language dataset from the Training dataset.

20 experiments across each model and language combination (i.e., five model comparisons for each language). Overall, we make several findings. First, we find that the Zero-Shot prompting style underperforms all other methods. Interestingly, adding the context information in the “Zero-Shot with Context” method improves the results. This suggests that including more information about the task (e.g., the definition of a euphemism) can improve performance.

Next, we can find that adding in-context examples in the “Few-Shot - Random Examples” and Few -Shot - Targeted Example” methods improves the “Zero-Shot with context” methods. Furthermore, we find that using Targeted examples universally improves performance over random examples. When we add more in-context examples, the performance continues to improve. For instance, “Few-Shot - Targeted Examples” improves from .801 with four in-context examples to .859 with eight examples. From a language-to-language perspective, we obtained the worst in Spanish, which is about 5% lower than the English results.

**Test Dataset Results.** The final competition results for our best system (i.e., Few Shot - Targeted Examples (8)) on the test dataset are shown in Table 3. The results indicate that the prompting with the English test cases performed substantially better than the prompting with the Spanish test cases, while the Chinese and Yorùbá test cases fell in between these two extremes. For the test experiments, the source of the sample cases to be included as random or

| Language | F1 | Precision | Recall |
|---|---|---|---|
| Chinese | .776 | .774 | .780 |
| English | .831 | .829 | .834 |
| Spanish | .598 | .622 | .659 |
| Yorùbá | .723 | .721 | .733 |

Table 3: F1, Precision, and Recall for each prompting technique for each language dataset from the Test dataset

targeted examples were pulled from the training datasets. The prompting proved most effective for the English dataset, and the results (F1=.831) were slightly higher than those measured during training. The results for both the Chinese (F1=.776) and the Yorùbá (F1=.723) datasets ended up falling between the "few shot random" and "few shot targeted (2)" prompt results for the training results for each language. The performance for the Spanish dataset fell (F1=.598) to only slightly better than the original "zero-shot" results.

When we look at the potential number of example cases to include with the targeted prompt, we find that with the English test cases, there was nearly 75% coverage. This means that 75% of the test PETs were also included in the training dataset. However, with the Spanish test cases, there was no overlap between the training data set and the test data set. The Chinese and Yorùbá data had test coverage between these two extremes. This may explain why the results with the Spanish dataset were so poor (0% coverage) and why the Chinese and Yorùbá datasets fell between random and targeted (some coverage).

**Error Analysis.** We analyzed a few of the errors to better understand how the model performed. For this analysis, we select one PET from the English dataset and one PET from the Chinese dataset.

In the English training dataset, the PET “disabled” showed good improvement by using the prompts. With the simple zero-shot prompt, all 16 examples were evaluated as being classified as a euphemism; however, seven of these examples were labeled as being literal in the ground-truth annotations. Adding context to the zero-prompt resulted in no improvement. Only slight improvement was realized when the few-shot prompt was used. However, with the few-shot prompt and eight examples, the evaluation matched 100%. The additional examples appeared to have given the model good context to discern between the nine euphemisms and seven literal cases. Overall, one potential cause for these findings is that certain terms, such as disabled, can appear in many contexts (euphemistic and not). The model is unable to understand which applies in a given context without strong examples. Other terms mostly used in euphemistic settings are easier for the system to detect.

In the Chinese training dataset, one of the PETs that showed improvement with each new prompt technique was the PET "环卫工人," which translates to "sanitation worker." GPT-4 sometimes translates this to "city beautician," which would be a euphemism. There are 30 examples in the training dataset, and each one is classified as a euphemism.

Only 5 of the 30 examples were included in the evaluation. With zero-shot prompting, all five failed to be classified as euphemisms. With each subsequent prompt technique, the performance improved to the last prompt, where four cases were identified correctly based on the label. This would indicate that the prompting added contextual data that influenced GPT-4. We believe that the term sanitation worker may not be a strong euphemism and needs substantial evidence from examples to change the prior of the model.

## 5 Future Work

While demonstrating the viability of our approach in identifying euphemisms, we also uncovered several research directions to pursue that could further enhance our understanding of the euphemistic speech capabilities of LLMs.

OpenAI's Chat GPT-4 model is a high-performing LLM trained on multi-lingual data. The LLM demonstrated its capability of translating the training datasets from the original language into English without additional fine-tuning. Limited testing during the development phase was performed using Mistral (Jiang et al.) and Llama-2 LLMs (Touvron et al.) but both exhibited zero-shot performance below Chat GPT-4. The main focus of the study was on improving performance using prompting strategies, so the team directed its efforts to refine the prompts. As highly capable LLM models are being released frequently, evaluating a variety of these models is an area of focus for future studies.

Our approach utilized only the model's inherent knowledge and a subset of the training data as additional knowledge to identify euphemisms. This additional knowledge was shown to significantly improve performance during the training phase. For the cases in which there were multiple samples to choose from, the current approach randomly selected the samples to include and the order they were listed. A future research direction is to determine if the selection of examples using those that are more closely related to the test case improves the performance. Also, does the order the samples are listed in the prompt affect the results?

When reviewing the test performance (Table 3), we noticed that not all languages performed comparably between training (Table 2) and test. When investigating the results for the lowest-performing dataset during the test phase (Spanish), we identified that no samples from the training dataset matched the PET in the test dataset. As noted, this additional knowledge was shown to be beneficial.

There are two approaches we could pursue to address this. One would be to locate additional datasets online or create datasets from open-source language repositories. A second approach would be to use a language model to generate the additional samples. The attraction to this approach is that we could generate samples of a new PET being used in a previously unseen manner and assist the model in recognizing the new usage of a phrase.

## 6 Conclusion

In this paper, we presented our approach for the 2024 FigLang Shared Task for multilingual Euphemism detection. We introduced a method using GPT-4 and in-context learning. This adjustment would be beneficial in a scenario in which the usage of a euphemism has changed over time, but the model has not yet been learned, or the model does not have a strong indication of being a euphemism without strong evidence. Future areas to research include 1) using the LLM to generate samples to include as examples to include in the multi-targeted prompt 2) improving the selection of targeted examples to identify those examples that are more closely related to the test case. 3) using the LLM to identify potential euphemisms from the text in question without being supplied with this information.

## ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation (NSF) under Grant No. 2145357.

## References


Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, et al. 2023. Mega: Multilingual evaluation of generative ai. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267.

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. Experience grounds language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8718–8735. Association for Computational Linguistics.

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, and Tom Henighan. Language models are few-shot learners.

Jaime G. Carbonell and Steven Minton. Metaphor and common-sense reasoning:.

Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. A computational approach to politeness with application to social factors.

Christian Felt and Ellen Riloff. Recognizing euphemisms and dysphemisms using sentiment analysis. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 136–145. Association for Computational Linguistics.

Martha Gavidia, Patrick Lee, Anna Feldman, and Jing Peng. CATs are fuzzy PETs: A corpus and analysis of potentially euphemistic terms.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b.

Sedrick Scott Keh. Exploring euphemism detection in few-shot and zero-shot settings.

Sedrick Scott Keh. 2022. Exploring euphemism detection in few-shot and zero-shot settings. In *Proceedings of the 3rd Workshop on Figurative Language Processing (FLP)*, pages 167–172.

Sedrick Scott Keh, Rohit Bharadwaj, Emmy Liu, Simone Tedeschi, Varun Gangal, and Roberto Navigli. 2022. EUREKA: EUphemism recognition enhanced through knn-based methods and augmentation. In *Proceedings of the 3rd Workshop on Figurative Language Processing (FLP)*, pages 111–117, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059. Association for Computational Linguistics.

Lingyao Li, Lizhou Fan, Shubham Atreja, and Libby Hemphill. 2023. "hot" chatgpt: The promise of chatgpt in detecting and discriminating hateful, offensive, and toxic comments on social media. *ACM Transactions on the Web*.

Moxin Li, Wenjie Wang, Fuli Feng, Yixin Cao, Jizhi Zhang, and Tat-Seng Chua. Robust prompt optimization for large language models against distribution shifts.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. 55(9):1–35.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Rijul Magu and Jiebo Luo. Determining code words in euphemistic hate speech using word embedding networks. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 93–100. Association for Computational Linguistics.

Abulimiti Maimaitituoheti, Yang Yong, and Fan Xiaochao. A prompt based approach for euphemism detection.

Abulimiti Maimaitituoheti, Yang Yong, and Fan Xiaochao. 2022. A prompt based approach for euphemism detection. In *Proceedings of the 3rd Workshop on Figurative Language Processing (FLP)*, pages 8–12.

Daniel Matter, Miriam Schirmer, Nir Grinberg, and Jürgen Pfeffer. 2024. Close to human-level agreement: Tracing journeys of violent speech in incel posts with gpt-4-enhanced annotations. *arXiv preprint arXiv:2401.02001*.

OpenAI. 2023. Chatgpt turbo 4 preview model 0125. `https://openai.com`. Accessed: 2024-03-02.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, C. J. Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. GPT-4 technical report.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback.

Pengcheng Qiu, Chaoyi Wu, Xiaoman Zhang, Weixiong Lin, Haicheng Wang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2024. Towards building multilingual language model for medicine. *arXiv preprint arXiv:2402.13963*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models.

Paul Trust, Kadusabe Provia, and Kizito Omala. 2022. Bayes at FigLang 2022 euphemism detection shared task: Cost-sensitive Bayesian fine-tuning and Venn-abers predictors for robust training under class skewed distributions. In *Proceedings of the 3rd Workshop on Figurative Language Processing (FLP)*, pages 94–99, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Xiaoyan Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, Ibrahim Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, and Michael Witbrock. Improving natural language inference using external knowledge in the science questions domain. 33(1):7208–7215.

Yuting Wang, Yiyi Liu, Ruqing Zhang, Yixing Fan, and Jiafeng Guo. 2022. Euphemism detection by transformers and relational graph attention network. In *Proceedings of the 3rd Workshop on Figurative Language Processing (FLP)*, pages 79–83.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models.

Peratham Wiriyathammabhum. 2022. Tedb system description to a shared task on euphemism detection 2022. In *Proceedings of the 3rd Workshop on Figurative Language Processing (FLP)*, pages 1–7.

Tianyu Wu, Shizhu He, Jingping Liu, Siqi Sun, Kang Liu, Qing-Long Han, and Yang Tang. A brief overview of ChatGPT: The history, status quo and potential future development. 10(5):1122–1136.

Wenxuan Zhang, Mahani Aljunied, Chang Gao, Yew Ken Chia, and Lidong Bing. 2024. M3exam: A multilingual, multimodal, multilevel benchmark for examining large language models. *Advances in Neural Information Processing Systems*, 36.

Wanzheng Zhu and Suma Bhat. Euphemistic phrase detection by masked language model. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 163–168. Association for Computational Linguistics.

Wanzheng Zhu and Suma Bhat. 2021. Euphemistic phrase detection by masked language model. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 163–168.