Mind the Biases: Quantifying Cognitive Biases in Language Model Prompting

Ruixi Lin and **Hwee Tou Ng**
Department of Computer Science
National University of Singapore
{ruixi,nght}@comp.nus.edu.sg

Abstract

We advocate the importance of exposing uncertainty on results of language model prompting which display bias modes resembling cognitive biases, and propose to help users grasp the level of uncertainty via simple quantifying metrics. Cognitive biases in the human decision making process can lead to flawed responses when we face uncertainty. Not surprisingly, we have seen biases in language models resembling cognitive biases as a result of training on biased text, raising dangers in downstream tasks that are centered around people's lives if users trust their results too much. In this work, we reveal two bias modes leveraging cognitive biases when we prompt BERT, accompanied by two bias metrics. On a drug-drug interaction extraction task, our bias measurements reveal an error pattern similar to the availability bias when the labels for training prompts are imbalanced, and show that a toning-down transformation of the drug-drug description in a prompt can elicit a bias similar to the framing effect, warning users to distrust when prompting language models for answers.[1]

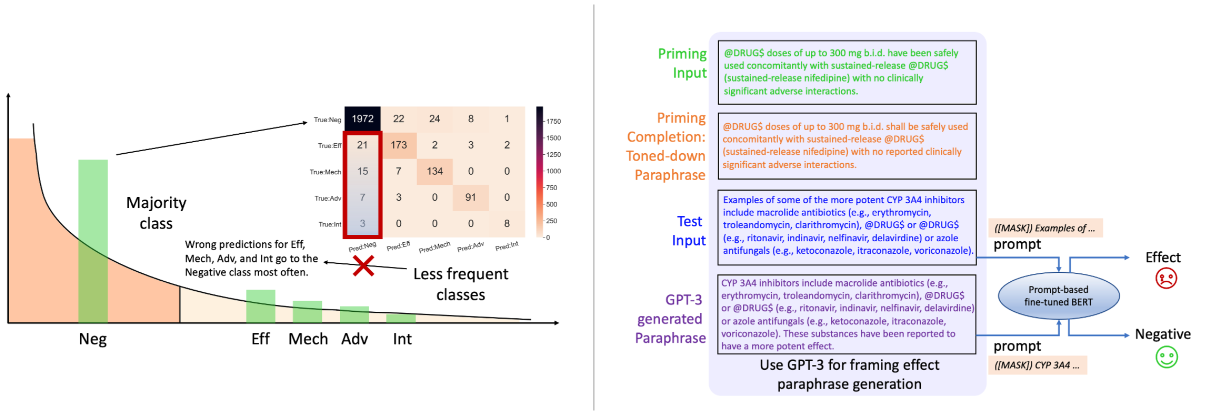
1 Introduction

Cognitive biases describe the flawed human response patterns for decision making under uncertainty (Tversky and Kahneman, 1974, 1981; Jacowitz and Kahneman, 1995; Kahneman and Frederick, 2002; Meyer, 2004). For example, when people are biased by the availability heuristic, they make probability judgments based on the ease with which information comes to mind (Tversky and Kahneman, 1973). Knowing cognitive biases can help predict what types of error will be made, which is also helpful for interpreting behaviors of generative systems such as language models, as they may err in a similar pattern as humans do, especially when the data used to build the systems carry man-made biases (Schwartz et al., 2022; Jones and Steinhardt, 2022). We are inspired by leveraging cognitive biases – systematic error patterns which deviate from rational decisions – to study error patterns of language models. We highlight the importance of exposing uncertainty to users of language models (Pinhanez, 2021), and leverage cognitive biases to quantify the level of imprecision in results when performing language model prompting via simple, perceptual metrics.

Some would argue that the biases in machines are a result of unmatched data distributions in training and test sets. However, merely matching training and test distributions does not solve the problem of biased predictions for long-tailed input distributions. For example, on the drug-drug interaction (DDI) dataset (Segura-Bedmar et al., 2013), the training and test distributions are identically skewed, and there are 100 times more *Negative* (non-interacting) drug pairs than the interacting drug pairs in both sets. Though performances on the development set and test set are not too bad for positive class inputs with a prompt-based BERT model (Devlin et al., 2019), the model still most frequently mistakes positive pairs for negative pairs, as shown by the confusion matrix in the left part of Figure 1. This label bias towards *Negative* mimics the availability bias. The availability bias is one of the most common cognitive biases in real life, especially in doctors' diagnoses which increase with years of training (Mamede et al., 2010; Saposnik et al., 2016). Moreover, an equal number of samples in each class during training does not guarantee that a "majority" class does not exist, especially when the input distribution of the negative class has a higher variance (i.e., highly diversified samples in the negative class) and the samples within the positive class share more common characteristics. Considering the input variance, even though sample sizes are the same, the negative class still can be viewed as the majority class. More on this can

[1] The source code of this paper is available at https://github.com/nusnlp/CBPrompt.

Findings of the Association for Computational Linguistics: ACL 2023, pages 5269–5281
July 9-14, 2023 ©2023 Association for Computational Linguistics



	Pred:Neg	Pred:Eff	Pred:Mech	Pred:Adv	Pred:Int
True:Neg	1972	22	24	8	1
True:Eff	21	173	2	3	2
True:Mech	15	7	134	0	0
True:Adv	7	3	0	91	0
True:Int	3	0	0	0	8

@DRUG$ doses of up to 300 mg b.i.d. have been safely used concomitantly with sustained-release @DRUG$ (sustained-release nifedipine) with no clinically significant adverse interactions.

@DRUG$ doses of up to 300 mg b.i.d. shall be safely used concomitantly with sustained-release @DRUG$ (sustained-release nifedipine) with no reported clinically significant adverse interactions.

Examples of some of the more potent CYP 3A4 inhibitors include macrolide antibiotics (e.g., erythromycin, troleandomycin, clarithromycin), @DRUG$ or @DRUG$ (e.g., ritonavir, indinavir, nelfinavir, delavirdine) or azole antifungals (e.g., ketoconazole, itraconazole, voriconazole).

CYP 3A4 inhibitors include macrolide antibiotics (e.g., erythromycin, troleandomycin, clarithromycin), @DRUG$ or @DRUG$ (e.g., ritonavir, indinavir, nelfinavir, delavirdine) or azole antifungals (e.g., ketoconazole, itraconazole, voriconazole). These substances have been reported to have a more potent effect.

Use GPT-3 for framing effect paraphrase generation

Figure 1: **Left**: An illustration of the negative class vs. the four positive classes of the DDI data. The confusion matrix of the development set predictions shows that the most frequent wrong prediction for positive input pairs is the negative class (the majority label), calling for the need to quantify the availability bias. **Right**: An observation on the framing effect paraphrase (in purple) and the original sentence (in blue): the paraphrase describes the two non-interacting drugs in a slightly more toned-down way compared to the original sentence, and it leads to the correct prediction whereas the original sentence does not, calling for the need to quantify the framing effect.

be found in Appendix A.

In addition to the availability bias, the framing effect is another common cognitive bias. The framing effect is prevalent in medical diagnoses (Loke and Tan, 1992), where doctors intentionally frame diagnoses positively ("90% chance to survive") or negatively ("10% chance to die") to make patients perceive the results differently. It was recently found that a failure mode of a code generation language model Codex (Chen et al., 2021) resembles the framing effect – how an input prompt is framed changes the model output predictably (Jones and Steinhardt, 2022). In our study, prompting BERT with paraphrases generated by toning-down the original inputs improves prediction results, suggesting a bias brought by the tone of the input sentences.

It is important to see that the biases found above are not the expected behavior of BERT as a prompt-based classification model, and our goal of this paper is to analyze these failure modes from the lens of cognitive biases and quantify them via simple metrics. We are devoted to warning practitioners about the risks of biased language model predictions, especially on biomedical tasks.

On a case study of the DDI extraction task, we measure output label distribution with content-free prompts and how model output changes when applying a toning-down transformation to prompt texts. Our key findings are:

- We have identified an error pattern similar to the availability bias when the labels for training prompts are imbalanced, and our measurements quantitatively show that the bias is highest towards the majority label.

- We have motivated a toning-down transformation of the drug-drug description in a prompt and found that this framing can elicit a bias similar to the framing effect.

2 Related Work

2.1 Cognitive Biases in Language Models

Recent work on studying behaviors of pretrained language models (PLMs) has revealed that some failure modes bear resemblance to cognitive biases. Wallace et al. (2019) study triggering prompts that fool the GPT-2 language model to generate contents that mimic hallucinations arising from cognitive biases. Zhao et al. (2021) find the majority label bias to be one of the pitfalls for GPT-3 (Brown et al., 2020), resembling the availability bias. Liu et al. (2022a) and Lu et al. (2022) show that specific order of training examples can lead to different model performance for GPT-3, analogous to the anchoring bias where estimates may be influenced by what information is provided first or last. Jones and Steinhardt (2022) capture failures in GPT-3 and Codex and find that error patterns of large language models (LLMs) resemble cognitive biases in humans. Agrawal et al. (2022) also find a bias in GPT-3 which is similar to the framing effect, where using separate prompts rather than a chained prompt leads to wrong answers for medication extraction. In a nutshell, most of these works focus on studying issues of LLMs and have discerned their error patterns' resemblance to human cognitive biases. We follow this line of research, and argue that relatively small PLMs, such as BERT, also display biases resembling human cognitive biases, and we propose metrics to quantify two of these biases.

2.2 Prompt-Based Language Models

As a booming research area, prompt-based methods show their success through few-shot learning performance for language models (Zhao et al., 2021; Jones and Steinhardt, 2022; Lu et al., 2022). However, prompts may not be understood by models the way humans do (Khashabi et al., 2022) and they affect biases in models (Webson and Pavlick, 2022; Utama et al., 2021; Prabhumoye et al., 2021). From a taxonomy viewpoint, prompt-based methods include: *Prompt design*, where the job is designing human-readable prompts to demonstrate to a frozen language model for downstream tasks (Brown et al., 2020); *Prompt tuning*, where tunable soft prompts are used for a frozen language model (Lester et al., 2021; Qin and Eisner, 2021; Sanh et al., 2022; Liu et al., 2022b); and *Prompt-based fine-tuning*, which utilizes fixed human-readable prompts to fine-tune a model (Scao and Rush, 2021; Gao et al., 2021; Schick and Schütze, 2021a,b; Tam et al., 2021), such as pattern-exploiting training (Schick and Schütze, 2021b; Tam et al., 2021). While the first two types are popular for large language models such as GPTs, prompt-based fine-tuning is more common when prompting BERT and other relatively small language models. In this work, we focus on prompt-based fine-tuning methods for BERT. Studies on interpretability focus on providing measures for the incompleteness that produces unquantified biases (Doshi-Velez and Kim, 2017). Here we aim to fill in the gap for quantifying the biases of prompt-based language models. In addition, adversarial input is a popular technique to interpret how a model is fooled, by tweaking image pixels (Akhtar and Mian, 2018; Li et al., 2019) or textual triggers (Wallace et al., 2019). However, in this work, we seek to study the effect of altered texts by leveraging cognitive bias patterns.

3 Proposed Metrics

We propose two metrics for quantifying the bias modes by the availability bias and the framing effect respectively in prompt-based BERT, helping users perceive how much bias comes with prompting results.

3.1 The Availability Bias Metric

The error by the availability bias can be viewed as a shortcut of how a model "thinks" an answer is easier to recall and occurs more readily than it actually occurs at test time, as long as it has seen many prompted instances of the same answer during training. On the DDI dataset, the majority label of prompts during training is *Negative* and the inference results show many false negatives. This resembles a situation when a human sees many negative examples, then the human inferences are more likely to be negative.

The Availability Bias Score. To quantify the availability bias for the DDI task, we are inspired by the work of (Zhao et al., 2021), where a language model's bias towards certain answers is estimated by feeding into the model a dummy test input that is content-free, i.e., with a dummy prompt, and measuring the deviation of the content-free prediction score from the uniform prediction score. Following this idea, we propose an availability bias metric via querying a model with multiple dummy test prompt inputs and computing the deviation of the prediction scores from the uniform prediction score as the bias measurement.

The intuition is that, when a dummy-content test prompt is given, the best that an unbiased model can do is to make a uniform random guess. If availability biases are present in the results, the number of predictions in each class will not be uniform. Henceforth, we can measure the deviation of the imbalanced predictions from the uniform prediction score to quantify the availability bias. We input dummy prompts to a language model, and measure the frequency of prediction of each class, and then compute the difference between class frequency and the uniform prediction score. For example, the DDI task features 5 classes, including 4 DDI types and *Negative*. Hence, the difference from $1/5 = 20\%$ is the availability bias score of each class.

In particular, we evaluate against a prompt-based fine-tuned BERT model and first obtain predictions conditioned on dummy prompt inputs. Let N denote the number of dummy test prompts, x_{dummy} denote a dummy prompt input.

$$\hat{y} = \arg\max_{y} p(y|x_{\text{dummy}}) \tag{1}$$

where $p(y|x_{\text{dummy}})$ is the softmax score obtained from the classification layer. Then we measure the frequency of each class prediction, i.e., the number of dummy predictions in each class (denoted by $count(C_i)$) divided by total number of dummy test prompts N.

$$count(C_i) = \sum_{j=1}^{N} \mathbb{1}\{\hat{y}_j = C_i\} \tag{2}$$

where $\mathbb{1}\{\cdot\}$ evaluates to 1 when the condition in the curly braces is met and 0 otherwise. Let M denote the number of classes. We propose the absolute deviation of the frequency from $1/M$ as the availability bias score for each class C_i, denoted by $Availability(C_i)$, and computed as follows:

$$Availability(Ci) = \left| \frac{count(C_i)}{N} - \frac{1}{M} \right| \tag{3}$$

For fairness in the dummy prompt design, we extract from each class an equal number of test instances and replace any UMLS keyword in the text with a dummy word, *N/A*, to form dummy prompts. The choice of dummy word follows the content-free prompt design in (Zhao et al., 2021). The reason to construct dummy prompts by extracting templates from each class is to mitigate the effect that a class-specific content-free input may correlate with surface class patterns. Moreover, our metric is robust to the number of dummy test prompts used, and we discuss it in Appendix B.

3.2 The Framing Effect Metric

The framing effect describes a biased perception about the same thing when it is framed differently, e.g., toning down an expression. We observe similar biases in BERT prompting when we transform the same input text describing a drug-drug interaction into a toned-down expression. When we use paraphrases as input for prompt-based fine-tuning and testing, the test predictions change and test F_1 score increases.

Measuring Framing Effect via Paraphrasing To measure the framing effect, we paraphrase the original drug-drug interaction descriptions to sound softer. We leverage the GPT-3 (Brown et al., 2020) model to build a paraphrase dataset, which contains 500 training instances, 50 development instances, and 300 test instances. To gauge the quality of paraphrase generation, we first compute BERTScore (Zhang et al., 2020) of the 850 generated sentences and their source reference sentences. BERTScore is a cosine similarity metric based on contextual embeddings to evaluate how much candidate and reference sentences match. The average BERTScore of all pairs is 97%, suggesting that the generated sentences are similar to the original sentences. However, BERTScore does not take into account the specific characteristics of a candidate, such as how toned-down a paraphrase is compared to the original sentence.

Therefore, we extend BERTScore and propose a Framing Effect Paraphrase (FEP) score to measure the framing effect-based P, R, F_1 scores for paraphrases and their source sentences. We focus on the framing effect of toning down a description and introduce a dictionary of toned-down words. The FEP score will award a paraphrase if any word in the paraphrase occurs in the dictionary, and penalize the source sentence if it already contains toned-down words. The reason to award a paraphrase is to encourage the use of toned-down words, and the source sentence is penalized because the best a paraphrase can do is to retain a tone-down word (since it is already in the source sentence), so the paraphrase will not receive a score for that word match. The dictionary of toned-down words, denoted as $\mathcal{A}$, is a list of toned-down words/rules, such as hedging words and uncertainty adjectives or adverbs, such as “may”, “can”, and “reportedly”, and words indicating conditions, such as “if” and “when”.

Given a source sentence x and a paraphrase $\hat{x}$, to compute precision, the FEP score not only computes a maximal matching similarity (by greedy matching) of each token $\hat{x_j}$ in $\hat{x}$ to a token in x, but also computes a reward score of each token in $\hat{x}$ by a scoring function $\phi_{\mathcal{A}}$, and precision is the larger of the two. Similarly, to compute recall, the FEP score computes both a matching similarity of each token x_i in x to a token in $\hat{x}$ and a penalty score of each token in x by $1 - \phi_{\mathcal{A}}(x_i)$, and recall is the smaller of the two. We then measure F_1 score by combining the precision and recall. The FEP precision, recall, and F_1 are denoted as $P_{\text{FEP}}, R_{\text{FEP}}, F_{\text{FEP}}$ respectively and are defined as follows:

$$P_{\text{FEP}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max(\max_{x_i \in x}(\mathbf{x}_i^{\top} \hat{\mathbf{x}}_j), \phi_{\mathcal{A}}(\hat{x}_j)) \tag{4}$$

where

$$\phi_{\mathcal{A}}(\hat{x}_j) = \begin{cases} 1 & \text{if } \hat{x}_j \in \mathcal{A} \\ 0 & \text{if } \hat{x}_j \notin \mathcal{A} \end{cases} \tag{5}$$

$$R_{\text{FEP}} = \frac{1}{|x|} \sum_{x_i \in x} \min(\max_{\hat{x}_j \in \hat{x}}(\mathbf{x}_i^\top \hat{\mathbf{x}}_j), 1 - \phi_{\mathcal{A}}(x_i)) \tag{6}$$

$$F_{\text{FEP}} = 2\frac{P_{\text{FEP}} \cdot R_{\text{FEP}}}{P_{\text{FEP}} + R_{\text{FEP}}} \tag{7}$$

The original sentence x and the paraphrase $\hat{x}$ are used as the input sentence of a prompt for fine-tuning BERT and testing, respectively. The prompt pattern will be introduced in Section 4.1. We then calculate conditional probabilities in a given F_{FEP} score range to measure the fine-grained performance changes caused by the toning down effect. For F_{FEP} in $[a, b)$, we compute the conditional probability of test pairs that are correctly predicted using the paraphrase input, given that the predictions of their original sentence are wrong. Specifically, we propose to measure the conditional probability, denoted as Δ in a given F_{FEP} score range, as follows:

$$\Delta = \frac{\sum_{k \in \mathcal{T}} \mathbb{1}\{f(x^k) \neq y^k, f(\hat{x}^k) = y^k\}}{\sum_{k \in \mathcal{T}} \mathbb{1}\{f(x^k) \neq y^k\}}, \tag{8}$$
$$\text{given} \quad F_{\text{FEP}}(x^k, \hat{x}^k) \text{ in } [a, b)$$

where $\mathcal{T}$ denotes the indices of test instances with F_{FEP} scores in the given range, f denotes the prompt-based language model, $f(x^k)$ and $f(\hat{x}^k)$ denote the model prediction for the k-th test input x^k and $\hat{x}^k$ respectively, and y denotes the correct label.

4 Experiments

4.1 Dataset and Model

We focus on the relation extraction task of drug-drug interactions, and use the DDIExtraction dataset (Segura-Bedmar et al., 2013) for our experiments. The DDI dataset was constructed with MedLine abstracts and DrugBank documents on drug-drug interactions. The DDI dataset uses 4 positive DDI types to annotate the semantic relation for the interaction of a drug pair, including *Mechanism* (DDI-mechanism), *Effect* (DDI-effect), *Advice* (DDI-advise), and *Int* (DDI-int), and a false class, which we refer to as the *Negative* class. *Mechanism* denotes the relation about a pharmacokinetic mechanism, *Effect* is used to annotate an effect or a pharmacodynamics mechanism, *Advice* is the relation describing an advice or recommendation regarding a drug interaction, and *Int* is the type for any other positive interaction types (Zhang et al., 2018). The classes are imbalanced with 85.2% *Negative*, 6.2% *Effect*, 4.9% *Mechanism*, 3.1% *Advice*, and 0.6% *Int*. Among all positive DDI types, *Mechanism* and *Advice* are better recognized, while *Effect* and *Int* are harder to be identified. For data preprocessing, we follow (Yasunaga et al., 2022) to replace the names of drugs of a pair to be classified with “@DRUG$”, and split the dataset into 25,296 training, 2,496 development, and 5,716 test instances.

The language model we study in this work is BERT-base[2], which uses a transformer (Vaswani et al., 2017) neural network pretrained on a 3.3 billion word corpus of general-domain English texts. For prompting BERT, we use the prompt-based fine-tuning method ADAPET[3] (Tam et al., 2021). The ADAPET method fine-tunes BERT via cloze-style prompt inputs with a [MASK] token (or tokens). The output is a softmax score produced by BERT over the [MASK] token vocabulary, which then corresponds to a class label. During training, the model is fine-tuned to minimize the sum of the decoupled label loss and the label-conditioned MLM loss. We stick to a single prompt pattern, “([MASK]) [TEXT]”, where [MASK] is the label phrase to be predicted and [TEXT] is the input drug pair description. The verbalizers are {“0": “false", “DDI-effect": “effect", “DDI-mechanism": “mechanism", “DDI-advise": “advice", “DDI-int": “interaction"}. We use a simple prompt pattern in this work. Since we obtain similar findings with more complex prompt patterns, we do not include them in this paper.

4.2 Measuring Availability Bias

In our experiments, we construct a total of 100 dummy test prompts, with 20 templates randomly extracted from each class. For the dummy test prompt design, we search for UMLS keyword contents in a sentence and replace them with dummy phrases *N/A*, and apply the prompt pattern: ([MASK]) [TEXT]. The [TEXT] part contains a

[2] https://huggingface.co/bert-base-uncased

[3] https://github.com/rrmenon10/ADAPET

	Availability bias score (%)				
Training size	10-shot	100-shot	1,000-shot	10,000-shot	25,296
Negative	26.3 (2.1)	77.7 (2.1)	39.7 (3.9)	47.0 (3.6)	52.0 (2.8)
Mechanism	20.0 (0.0)	20.0 (0.0)	13.7 (0.5)	17.3 (1.2)	16.7 (1.3)
Advice	20.0 (0.0)	18.3 (1.7)	8.3 (2.6)	7.0 (2.4)	8.3 (2.6)
Effect	33.7 (2.1)	20.0 (0.0)	16.3 (2.1)	12.7 (2.5)	11.7 (2.4)
Int	20.0 (0.0)	19.3 (0.5)	1.3 (0.5)	10.0 (0.8)	15.3 (1.2)

Table 1: Availability bias score (%) for each class on the DDI task. Column 2-5: few-shot training settings. Column 6: full training set. Mean and standard deviation are shown.

Test data	**# pairs**	F_1
Paraphrase, including invalid paraphrases	300	44.6
Original sentences of the above	300	10.2
Paraphrase, excluding invalid paraphrases	208	55.7
Original sentences of the above	208	9.0

Table 2: F_1 score (%) on the framing effect test set.

test sentence with multiple *N/A*s and the [MASK] part will be the predicted label during testing. For UMLS keyword extraction, we exploit MetaMap[4] and its Python wrapper[5]. An example dummy test prompt is shown below.

```
([MASK]) @DRUG$ competes with a N/A
of N/A for N/A N/A N/A notably
N/A N/A N/A N/A N/A @DRUG$
N/A N/A N/A and N/A N/A
```

The language model we measure against is BERT-base, fine-tuned via the ADAPET method with prompt inputs of the full DDI training set and few-shot training sets including 10, 100, 1,000, 10,000-shot training settings. Note that the original test F_1 scores of the positive DDI types on the 10, 100, 1,000, 10,000-shot, and full training set are 5.04%, 12.36%, 56.16%, 74.64%, and 80.36% respectively. We repeat the experiments three times with different random seeds, and report the mean and standard deviation.

Table 1 shows the availability bias score (%) for each class, on different fine-tuned BERT models. The rightmost column in Table 1 represents the scores for the BERT model fine-tuned on the full training set. The upper limit for the availability bias score is (100-20)/100=80%, and the closer the bias score gets to the upper limit, the more biased the model makes predictions towards the associated class. As expected, the bias towards the *Negative* class is the largest, by 52%, suggesting that when supposedly making random guess for dummy inputs, the model's behavior is vastly biased towards predicting drug pairs as no relation.

In addition, results in column 2 to column 5 in Table 1 present availability bias scores for few-shot training cases. It is interesting to see that the 10-shot trained model exhibits lower bias score towards the majority class. However, its accuracy on the original full test set is only 5.04%. For the remaining cases, the conclusion that the model outputs are biased towards the majority class also holds in few-shot training settings.

Though one may argue that labels in prompts do not matter much for classification as in traditional supervised learning (Min et al., 2022), we find that it is not true from our availability bias scores obtained. The label in a prompt still plays an important part in prompt-based training, leading to availability bias-like predictions. The practical implication of knowing this bias pattern is that when users see model predictions, they can be informed that a prediction given by a model is biased towards the predicted label by the quantified amount.

4.3 Measuring Framing Effect

We first build the paraphrase dataset, where we randomly select 500 training instances, 50 development instances, and 300 test instances from the full DDI training, development, and

[4] https://lhncbc.nlm.nih.gov/ii/tools/MetaMap.html

[5] https://github.com/AnthonyMRios/pymetamap

F_{FEP}	**# Ori. wrong**	**# Pp. correct**	Δ
[0.99, 1.00)	78	77	98.7
[0.97, 0.99)	23	12	52.2
[0.95, 0.97)	35	27	77.1
[0.00, 1.00)	141	119	84.4

Table 3: Framing effect score: conditional probabilities (Δ) on the DDI test set. # **Ori. wrong** denotes the number of wrong predictions with the original inputs. # **Pp. correct** denotes the number of correct predictions using paraphrase inputs within those that are originally predicted wrongly. Results for F_{FEP} scores lower than 0.95 are not presented and used for analysis as there are too few such test instances (less than 3).

test set respectively for paraphrasing. The paraphrases are generated by prompting GPT-3 with a demonstration and the actual query, where a priming example (in blue) is appended to the test sentence to be paraphrased (denoted as [INPUT]). In our experiments, we design 8 priming examples and randomly pick one of them as demonstration. An example GPT-3 query is given below.

> Paraphrase the following drug interaction description. === Although @DRUG$ exerts a slight intrinsic anticonvulsant effect, its abrupt suppression of the protective effect of a @DRUG$ agonist can give rise to convulsions in epileptic patients. Description: @DRUG$ exerts a slight intrinsic anticonvulsant effect, and its abrupt suppression of the protective effect of a @DRUG$ agonist is reportedly to give rise to convulsions in epileptic patients. === [INPUT] Rephrase the above description to sound soft. Write the description in a warm tone. Description:

We illustrate several GPT-3 generated paraphrases of the test instances in Figure 2. For training and testing, we use all the generated paraphrases, although some paraphrases contain hallucinations (e.g., an untruthful trailing sentence that may come from the priming example) or miss major content (e.g., missing the mention of a drug to be predicted). The language model we measure against is the BERT-base model, fine-tuned via the ADAPET method on the 500 training instances. An example of a prompt input to BERT is as follows:

> ([MASK]) If you are taking @DRUG$ or other potent CYP3A4 inhibitors such as other azole antifungals (eg, itraconazole, @DRUG$) or macrolide antibiotics (eg, erythromycin, clarithromycin) or cyclosporine or vinblastine, the recommended dose of DETROL LA is 2 mg daily.

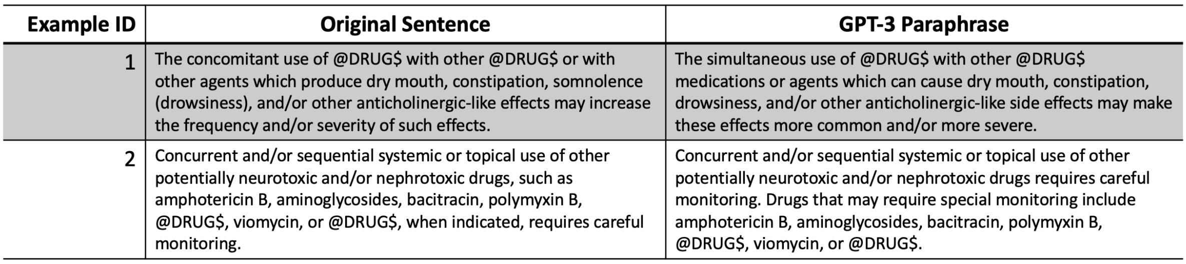
Table 2 shows the test F_1 scores on both the original test sentences and their GPT-3 paraphrases. As shown by the last two rows, the 208 valid paraphrases obtain an F_1 score of 55.7%, which is 46.7% higher than the 208 original sentences which obtain an F_1 score of 9.0%.

More importantly, for the 208 drug pairs with valid paraphrases, we show in Table 3 that the Δ is 84.4%, and if we focus on highly toned-down paraphrases in F_{FEP} range $[0.99, 1.00)$, the conditional probability reaches 98.7%, showing that framing an original drug-drug interaction description into a toned-down paraphrase helps to improve relation extraction. These results suggest that toning down the input text in a prompt can elicit a bias in predictions qualitatively similar to the framing effect.

Furthermore, we illustrate the original sentences and their framed paraphrases through some test pairs in Figure 2. In Example 1, the correct relation "effect" is identified given the paraphrase input, while no interaction is detected given the original sentence input. Compared to the original text which uses the word "produce" to describe side effects, the words "can cause " used in the paraphrase are more toned-down. In Example 2, the correct relation is no interaction, which is identified correctly using the paraphrase input, while the wrong prediction "effect" is made using the original sentence. In the original sentence, "requires" is used for the list of drugs, while "may require" is used in the paraphrase, toning down the expression.

5 Discussion

Few-shot Training vs. the Availability Bias. We have seen from Table 1 that at 10-shot, the availability bias towards *Negative* is not as obvious and the scores are more similar among the five classes. This is in contrast to the other few-shot learning cases with more training instances, where the availability bias becomes more obvious for the negative



Example ID	Original Sentence	GPT-3 Paraphrase
1	The concomitant use of @DRUG$ with other @DRUG$ or with other agents which produce dry mouth, constipation, somnolence (drowsiness), and/or other anticholinergic-like effects may increase the frequency and/or severity of such effects.	The simultaneous use of @DRUG$ with other @DRUG$ medications or agents which can cause dry mouth, constipation, drowsiness, and/or other anticholinergic-like side effects may make these effects more common and/or more severe.
2	Concurrent and/or sequential systemic or topical use of other potentially neurotoxic and/or nephrotoxic drugs, such as amphotericin B, aminoglycosides, bacitracin, polymyxin B, @DRUG$, viomycin, or @DRUG$, when indicated, requires careful monitoring.	Concurrent and/or sequential systemic or topical use of other potentially neurotoxic and/or nephrotoxic drugs requires careful monitoring. Drugs that may require special monitoring include amphotericin B, aminoglycosides, bacitracin, polymyxin B, @DRUG$, viomycin, or @DRUG$.

Figure 2: Examples of GPT-3 generated paraphrases of the test instances. F_{FEP} (toned-down paraphrasing scores) from top to bottom: 0.994, 0.988.

class as the number of training instances increases. It does not suggest that training on more instances will worsen the availability bias, but more class-biased training prompts will amplify the availability bias. That is, since more negative class instances are drawn for a larger number of training instances, the majority class has been seen by the model more frequently, causing biased predictions due to this increased availability. Prompt-based learning is not immune to imbalanced class distribution even under few-shot settings, as it is sometimes hard to obtain real class-balanced few-shot instances (this is elaborated in Appendix A).

6 Conclusion

In this work, we identify and quantify two bias modes in BERT's prompt-based predictions, leveraging the availability bias and the framing effect on biomedical drug-drug interaction extraction. The error mode of the availability bias suggests that the label for a prompt still matters for prompt-based learning, as shown by a large availability bias score towards the majority class, which is 52% on a scale of 0 to 80%. We also find that a toning-down transformation of the drug-drug description in a prompt can elicit a bias similar to the framing effect, since when we tone down the input description, 84.4% of drug pairs that are wrongly classified with the original text are now correctly predicted with their toned-down paraphrases. For highly toned-down paraphrases (as measured by F_{FEP} above 0.99), this conditional probability reaches 98.7%. The magnitude of these biases suggests that language model users need to be aware of the imprecision of their prompting results.

7 Limitations

The limitations are that our use of GPT-3 sometimes generates hallucinated texts, thus reducing the effectiveness in generating valid paraphrases. The dictionary of toned-down words could include more semantic rules or could be built automatically, which will be left as future work.

References

Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, and David Sontag. 2022. Large language models are zero-shot clinical information extractors. *arXiv preprint arXiv:2205.12689*.

Naveed Akhtar and Ajmal Mian. 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. *arXiv preprint arXiv:1801.00553*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya

Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830.

Karen E. Jacowitz and Daniel Kahneman. 1995. Measures of anchoring in estimation tasks. *Personality and Social Psychology Bulletin*, 21:1161–1166.

Erik Jones and Jacob Steinhardt. 2022. Capturing failures of large language models via human cognitive biases. In *Advances in Neural Information Processing Systems*.

Daniel Kahneman and Shane Frederick. 2002. Representativeness revisited: Attribute substitution in intuitive judgment. *Heuristics and biases: The psychology of intuitive judgment*, 49:49–81.

Daniel Khashabi, Xinxi Lyu, Sewon Min, Lianhui Qin, Kyle Richardson, Sean Welleck, Hannaneh Hajishirzi, Tushar Khot, Ashish Sabharwal, Sameer Singh, and Yejin Choi. 2022. Prompt waywardness: The curious case of discretized interpretation of continuous prompts. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3631–3643.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. 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.

Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. 2019. Certified adversarial robustness with additive noise. In *Advances in Neural Information Processing Systems*, pages 9464–9474.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022a. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022b. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68.

Wing Hong Loke and Kai Foong Tan. 1992. Effects of framing and missing information in expert and novice judgment. *Bulletin of the Psychonomic Society*, 30:187–190.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 8086–8098.

Sílvia Mamede, Tamara van Gog, Kees van den Berge, Remy M. J. P. Rikers, Jan L. C. M. van Saase, Coen van Guldener, and Henk G. Schmidt. 2010. Effect of availability bias and reflective reasoning on diagnostic accuracy among internal medicine residents. *Journal of the American Medical Association*, 304:1198–1203.

David E. Meyer. 2004. Semantic priming well established. *Science*, 345:523–523.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Claudio S. Pinhanez. 2021. Expose uncertainty, instill distrust, avoid explanations: Towards ethical guidelines for AI. *CoRR*, abs/2112.01281.

Shrimai Prabhumoye, Rafal Kocielnik, Mohammad Shoeybi, Anima Anandkumar, and Bryan Catanzaro. 2021. Few-shot instruction prompts for pretrained language models to detect social biases. *arXiv preprint arXiv:2112.07868*.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Gustavo Saposnik, Donald Redelmeier, Christian C Ruff, and Philippe N Tobler. 2016. Cognitive biases associated with medical decisions: A systematic review. *BMC Medical Informatics and Decision Making*, 16.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 255–269.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352.

Reva Schwartz, Apostol Vassilev, Kristen K. Greene, Lori Perine, Andrew Burt, and Patrick Hall. 2022. Towards a standard for identifying and managing bias in artificial intelligence. *Special Publication (NIST SP)*.

Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo. 2013. Semeval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350.

Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and simplifying pattern exploiting training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4980–4991.

Amos Tversky and Daniel Kahneman. 1973. Availability: A heuristic for judging frequency and probability. *Cognitive Psychology*, 5:207–232.

Amos Tversky and Daniel Kahneman. 1974. Judgment under uncertainty: Heuristics and biases. *Science*, 185:1124–1131.

Amos Tversky and Daniel Kahneman. 1981. The framing of decisions and the psychology of choice. *Science*, 211:453–458.

Prasetya Utama, Nafise Sadat Moosavi, Victor Sanh, and Iryna Gurevych. 2021. Avoiding inference heuristics in few-shot prompt-based finetunings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9063–9074.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2153–2162.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

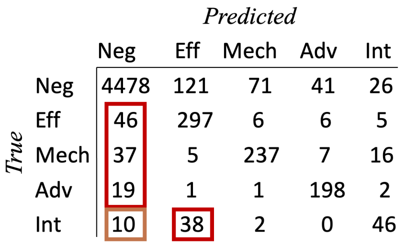
Yijia Zhang, Wei Zheng, Hongfei Lin, Jian Wang, Zhihao Yang, and Michel Dumontier. 2018. Drug–drug interaction extraction via hierarchical rnns on sequence and shortest dependency paths. *Bioinformatics*, 34:828–835.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*.

Appendix

A The Majority Label in Training

Since the availability bias arises from the majority label in training, where the majority label is typically defined by class size, one may argue that there should not be an availability bias with an equal number of class instances. However, we constructed a balanced training data set of equal class size for fine-tuning (by sampling 2,000 instances from each of the five classes, and all four positive classes include duplicate instances since their class sizes are less than 2,000). Interestingly, we still observe that the predictions for the positive classes are biased towards the negative class on the test set, as shown by the confusion matrix in Figure 3. Except *Int*, wrong predictions most frequently fall into the negative class for all other positive classes *Effect*, *Mechanism*, *Advice*. This does not contradict our conclusion that the availability bias exists, and it further suggests that the majority label should not be solely defined by class size, but the class with the highest input variance.



		Predicted				
		Neg	Eff	Mech	Adv	Int
True	Neg	4478	121	71	41	26
	Eff	46	297	6	6	5
	Mech	37	5	237	7	16
	Adv	19	1	1	198	2
	Int	10	38	2	0	46

Figure 3: Test confusion matrix for class-balanced training data (2,000 instances per class). Balanced training data does not alleviate the availability bias at inference time. For positive classes except *Int*, wrong predictions most frequently fall into the negative class.

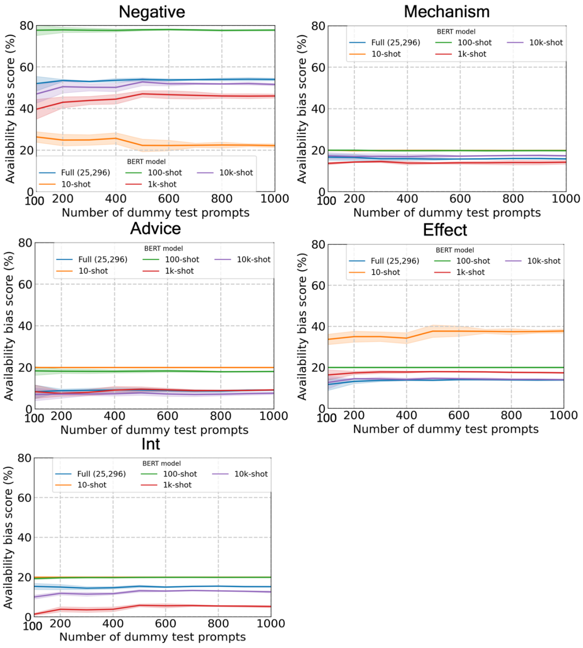


Figure 4: Number of dummy test prompts for availability bias measurement.

B Number of Dummy Test Prompts for Availability Bias Measurement

We increase the number of dummy test prompts N to show the stability of our availability bias metric, where N ranges from 100 to 1000 with a step size of 100. We repeat our experiments three times for each N and calculate the mean and standard deviation. When creating dummy test prompts, if the number of dummy templates that need to be drawn from a class exceeds the class size, we enable upsampling of duplicate templates from that class. Figure 4 shows that the availability bias measurement is stable for $N \geq 100$, suggesting that our proposed metric can be used with as few as 100 dummy test prompts.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 7

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract, Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4, we use pretrained BERT and build upon the ADAPET open source code for experiments Section 4.2, we use the MetaMap software for UMLS keywords extraction Section 4.3, we use GPT-3 for paraphrase generation

☑ B1. Did you cite the creators of artifacts you used?
Section 3, Section 4

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Not applicable. Left blank.

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4.1

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4.1

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4.2, 4.3

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.