Evaluating the fairness of task-adaptive pretraining on test data for few-shot text classification

Anonymous ACL submission

Abstract

Few-shot learning benchmarks are critical for evaluating modern NLP techniques. But it's possible that benchmarks favor methods which easily make use of unlabeled text, because researchers can pretrain their models on unlabeled text from the test set. Given the dearth of research on this potential problem, we run experiments to quantify the bias caused by pretraining on unlabeled test set text instead of on independent text. A controlled experiment on 25 classification tasks and 2 language models—BERT and GPT-2—demonstrates that there's negligible overoptimism when the training set contains 100 texts and the test set contains 200 texts, and no overoptimism when the test set contains 500 texts. Furthermore, we demonstrate the importance of repeated subsampling when studying few-shot text classification, and recommend that few-shot learning benchmarks include multiple training folds. Code and data are available here: https://github.com (currently omitted for anonymity).

1 Introduction

For NLP benchmarks, it's standard to release text from the test set. This allows researchers to submit a file of predictions instead of submitting code. A potential concern is that researchers can technically use this text during training. Consider the Real-world Annotated Few-shot Tasks (RAFT) benchmark (Alex et al., 2021), which contains "few-shot" text classification tasks—tasks where the training set contains a relatively small number of labeled examples. Below is an excerpt from the RAFT paper (emphasis added):

> For each task, we release a public training set with 50 examples and a larger unlabeled test set. *We encourage unsupervised pre-training on the unlabelled examples* and open-domain information retrieval.

In the RAFT competition, a model is evaluated by scoring its predictions on the same set of unlabeled text which the model may have been trained on (using an unsupervised training procedure).

It's wrong to train on test set features with their labels and then evaluate a model on the test set when one needs to estimate a model's performance on unseen data. Test set performance would be overoptimistic (Hastie et al., 2009). This fact is widely known. But what if, as encouraged by Alex et al. (2021), a model is trained on test set features *without* test set labels? This paper studies this question for the domain of few-shot text classification.

2 Motivation

NLP benchmarks for few-shot learning are widespread, as having only a handful of labeled examples is more common in practice. One consideration when designing these benchmarks is that some few-shot approaches can—at least theoretically—use unlabeled text from the test set. With Pattern-Exploiting Training (Schick and Schütze, 2021), for example, one can train the final classifier on test set text with soft predictions made from an ensemble of supervised models. Or, with Pre-trained Prompt Tuning (Gu et al., 2022), one can pretrain the language model (LM) on test set text before prompt-tuning on the labeled training set. A more classical approach would be to train a word2vec model (Mikolov et al., 2013) on unlabeled test set text, run this model on training text to get embeddings, and finally train a classifier on these embeddings with labels from the training set.

While the ability to exploit unlabeled text is useful, applying this ability to test set text could be substantively different than applying it to text which is statistically independent of the test set. This difference in methodology is more concerning in the few-shot setting than in the many-shot setting. It's conceivable that differences between few-shot

methods are as attributable to differences in how unlabeled text is used than how the few, labeled examples are used. This begs the question: in few-shot text classification benchmarks, can there theoretically be a bias favoring methods which exploited unlabeled text from the test set?

As indicated by the quote in §1, the RAFT benchmark implicitly assumes that the answer is no. It is not a fringe opinion that test set features may be used. The popular textbook by Hastie et al. (2009) contains the following passage without a reference or evidence[1] (emphasis added):

> There is one qualification: *initial unsupervised screening steps can be done before samples are left out.* For example, we could select the 1000 predictors with highest variance across all 50 samples, before starting cross-validation. *Since this filtering does not involve the class labels, it does not give the predictors an unfair advantage.*

The opposite opinion—that exploiting unlabeled test set features is unfair—may be more popular. For example, Gururangan et al. (2020) contains the following criticism of another study when comparing performances on a popular text classification benchmark:

> Thongtan and Phienthrakul (2019) report a higher number (97.42) on IMDB, but they train their word vectors on the test set.

We also conducted an informal, online poll of the `MachineLearning` subreddit. Among 36 users who had an opinion on this question, 35 believed that pretraining on unlabeled test set text before classification is usually or always unfair.[2]

3 Related work

Moscovich and Rosset (2022) contains experiments and theory for unsupervised methods which are common to tasks involving tabular data. They find that estimators of out-of-sample performance which were subject to these methods may be biased positively or negatively, depending on all of the parameters of the problem.

[1] In an email correspondence with Hastie regarding this passage, Hastie clarified that: "This is an interesting topic which we have explored with our students in various ways, but I personally am still pretty happy with the statement in ESL, although not as adamant as I was at the time we wrote it."

[2] `urlCurrentlyOmittedforAnonymity`

4 Experiment design

In the absence of theory or experiments in NLP, this paper studies how much pretraining on unlabeled test set text biases test set performance for 25 diverse text classification tasks and two types of LMs: BERT (Devlin et al., 2019), and GPT-2 (Radford et al., 2019). Descriptions of the 25 classification tasks are included in Appendix A. The number of classes in each task ranges from 2 to 18.

At a high level, the goal of the experiment is to first establish that pretraining is beneficial, in line with Gururangan et al. (2020). Second, given that pretraining has a detectable benefit, the experiment measures the accuracy difference between using test set text for the pretraining stage—an arguably unfair methodology—versus using text which is independent of the test set—an inarguably fair methodology.

In more detail, the experiment starts by drawing three subsamples (without replacement) from the full sample of data for a given text classification task:

- `extra`: n observations which are optionally used for pretraining
- `train`: 100 observations for supervised classification training
- `test`: n (either 200 or 500) observations which will be used to report accuracy.

Next, three accuracy estimators are computed. The procedures used to obtain them are described below.

4.1 $\text{acc}_{\text{extra}}$

1. Train a freshly loaded, pretrained LM on the n unlabeled texts in `extra` using the LM's pretraining objective.
2. Add a linear layer to this model, and finetune all of the LM's weights to minimize classification cross entropy loss on `train`.
3. Compute the classification accuracy of this model on `test`.

Step 1 is task-adaptive pretraining—a procedure broadly recommended by Gururangan et al. (2020). Step 2 is the canonical way of training a transformer-based LM for a classification task, according to Section 2 of Zhang et al. (2021).

$\text{acc}_{\text{extra}}$ is clearly an unbiased estimator of out-of-sample accuracy, because it never trains on data from `test`.

4.2 acc_{test}

acc_{test} is identical to $\text{acc}_{\text{extra}}$, except that pretraining is done on `test` instead of `extra` in step 1.

acc_{test} represents what one might see in a competition like RAFT, where pretraining on `test` is encouraged. It's unclear whether this accuracy estimator is unbiased, because it was (pre)trained and evaluated on the same set of test set text. A reasonable hypothesis is that it's optimistic, i.e., $\text{E}[\text{acc}_{\text{test}}] > \text{E}[\text{acc}_{\text{extra}}]$ = out-of-sample accuracy.

4.3 acc_{base}

acc_{base} doesn't do pretraining; it doesn't make any use of unlabeled text. It simply trains a pretrained LM on `train` to do classification, and then computes this model's accuracy on `test`.

This score is a control. If there's no boost going from acc_{base} to $\text{acc}_{\text{extra}}$, then it shouldn't be surprising that there's no boost going from $\text{acc}_{\text{extra}}$ to acc_{test}.

4.4 Subsampling

The three accuracy estimators are paired, because their classification training and test sets are identical. The only difference is the source of unlabeled text for pretraining. For $\text{acc}_{\text{extra}}$, the source is independent of test set text. For acc_{test}, the source is exactly the test set text. For acc_{base}, no unlabeled text is used.

A potentially important source of variation in this experiment is the particular subsamples, i.e., the particular realizations of `extra`, `train`, and `test` for a given classification task. To expose this variation, the experiment procedure is repeated 50 times for each classification task for $n = 200$, and 20 times for $n = 500$. In other words, for $n = 200$, and for each of the 25 classification tasks, 50 ($\text{acc}_{\text{extra}}$, acc_{test}, acc_{base}) triples are computed.

Appendix B explains more experiment choices.

5 Results

Appendix D.2 contains visualizations of the distributions of the paired differences: $\text{acc}_{\text{extra}}$ - acc_{base} and acc_{test} - $\text{acc}_{\text{extra}}$. $\text{acc}_{\text{extra}}$ - acc_{base} is a control: it's the accuracy boost from pretraining on independent text versus not pretraining at all. acc_{test} - $\text{acc}_{\text{extra}}$ is the main quantity of interest: it's the accuracy boost from pretraining on test set text instead of on independent text, i.e., it's the evaluation bias.

	BERT	GPT-2
$n = 200$	0.041 / 0.0033	0.062 / -0.0001
$n = 500$	0.038 / 0.0008	0.039 / -0.0021

Table 1: Sample means of paired differences between accuracies taken across all subsamples of the 25 text classification tasks. For each cell, the upper-left of the diagonal corresponds to the sample mean of $\text{acc}_{\text{extra}}$ - acc_{base}, and the lower-right corresponds to the sample mean of acc_{test} - $\text{acc}_{\text{extra}}$.

Table 1 contains means of the paired differences for each configuration of the experiment. This table, along with the visualizations in Appendix D.2, roughly suggest that while pretraining is consistently beneficial, pretraining on test set text does not significantly bias test set performance.

A more complete analysis of this data is motivated and performed in the next section.

6 Analysis

Reporting means is not enough, especially when studying few-shot learning. The two figures in Appendix D.2 demonstrate that there is considerable variance, despite pairing.[3] While these visualizations tell us about how raw accuracy differences vary, they do not tell us how the mean accuracy difference varies. We seek a neat answer to the core questions: on our benchmark of 25 classification tasks, how much does the average performance differ between two modeling techniques, and how much does this average difference vary?

One way to communicate the variance is to estimate the standard error of the mean difference across classification tasks. But the standard error statistic can be difficult to interpret (Morey et al., 2016). Furthermore, its computation is not completely trivial due to the data's hierarchical dependency structure: each triple, ($\text{acc}_{\text{extra}}$, acc_{test}, acc_{base}), is drawn from (`train`, `test`), which is itself drawn from the given classification dataset.

6.1 Model

This analysis does not aim to estimate standard errors. Instead, posterior distributions will be estimated by fitting a hierarchical model:

[3] One source of variance is intentionally introduced: the subsamples/splits, as explained in §4.4. The other source of variance is inherent: the added linear layer to perform classification is initialized with random weights.

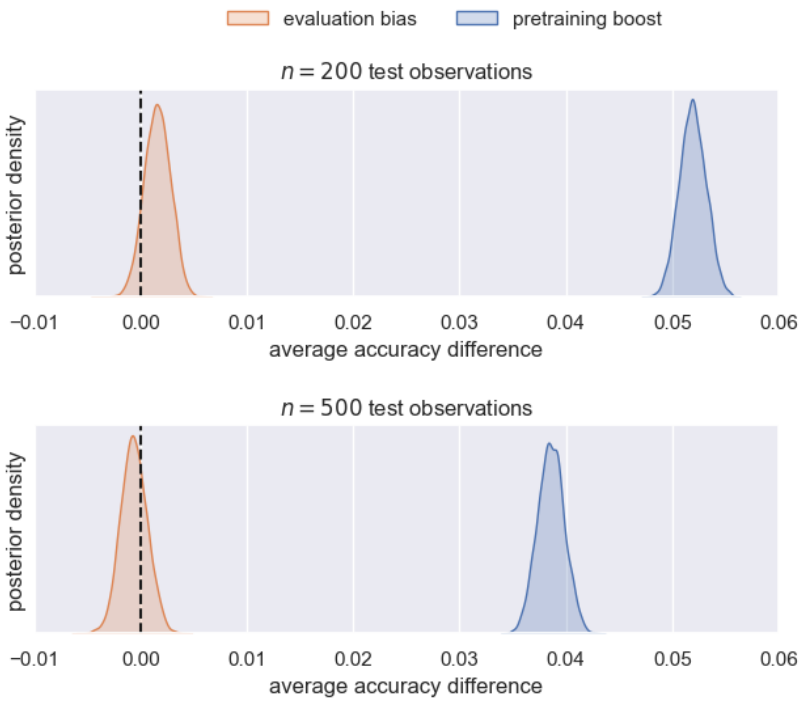


Figure 1: Distributions of average accuracy differences for $n = 200$ test observations (top) and $n = 500$ (bottom). Each distribution is estimated by sampling from the posterior predictive distribution of $Y_{ijk1} - Y_{ijk0}$ (6.1). The evaluation bias is akin to acc_{test} - $\text{acc}_{\text{extra}}$. The pretraining boost is akin to $\text{acc}_{\text{extra}}$ - acc_{base}. Each difference is a marginal effect—averaged across 2 LM types (BERT and GPT-2), the 25 classification tasks, and their subsamples.

$$Y_{ijkl} \sim \text{Binomial}(n, \lambda_{ijkl}) \quad (1)$$

$$\text{logit}(\lambda_{ijkl}) = \mu + \alpha z_i + U_j + V_{jk} + \beta x_{ijkl} \quad (2)$$

$$\mu \sim \text{Normal}(0, 1) \quad (3)$$

$$\alpha \sim \text{Normal}(0, 5) \quad (4)$$

$$U_j \sim \text{Normal}(0, \sigma_U) \quad (5)$$

$$V_{jk} \sim \text{Normal}(0, \sigma_V) \quad (6)$$

$$\beta \sim \text{Normal}(0, 1) \quad (7)$$

$$\sigma_U, \sigma_V \sim \text{HalfNormal}(0, 1) \quad (8)$$

(1) number of correct predictions

(2) logit link, additive effects

(3) prior for the global intercept

(4) prior for the effect of the type of LM (BERT or GPT-2)—a control variable

(5) prior for the effect of the classification task (partial-pooled to reduce overfitting)

(6) prior for the nested effect of the task's subsampled dataset

(7) prior for the effect of interest ($x_{ijk1} = 1$ indicates the modeling intervention)

(8) prior for standard deviations.

The model is fit using Markov Chain Monte Carlo, using the interface provided by the `bambi` package (Capretto et al., 2022). 4,000 samples from the posterior were drawn for each effect. Appendix E includes a simulation that demonstrates the model's ability to correctly recover null and non-null effects.

6.2 200 test observations

Figure 1 (top-right distribution) validates that there's a boost from pretraining on independent text versus not pretraining at all. Pretraining works, so there may be a detectable evaluation bias.

Figure 1 (top-left distribution) provides evidence that pretraining on test set text (instead of on independent text) causes overoptimism; much of the probability mass is on the positive side. But the magnitude of the bias is somewhat negligible. To put the accuracy difference statistic in context, the average gap between adjacent submissions in the RAFT leaderboard is 0.006. The 25$^{\text{th}}$ percentile of submission gaps is 0.002, so it's not unlikely for some ranks to switch by having one model pretrain on the test set (with $n = 200$) while another, equally performant model pretrains on independent text.

6.3 500 test observations

While $n = 200$ is not uncommon in real-world few-shot problems, most benchmarks contain larger test sets. Moscovich and Rosset (2022) found that the evaluation bias caused by certain unsupervised methods for tabular data tends to vanish in the limit of n. How does the small but evident bias found for $n = 200$ change when n is increased to 500? A reasonable hypothesis is that the bias gets even closer to 0.

Figure 1 (bottom-right distribution) sanity checks that there's a significant pretraining boost to detect. Figure 1 (bottom-left distribution) shows that the bias now hovers tightly around 0, validating the result from Moscovich and Rosset (2022).

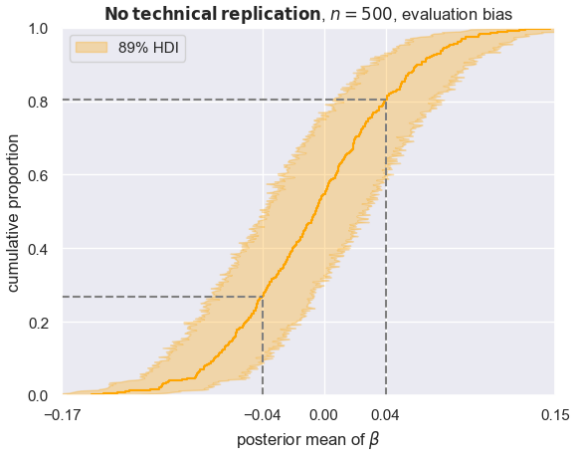


Figure 2: evaluation bias is akin to acc_{test} - $\text{acc}_{\text{extra}}$.

7 Meta-analysis

§4.4 briefly argues for subsampling multiple datasets from the full classification dataset. To assess this argument, the analysis was repeated on 500 random slices of the $n = 500$ dataset of accuracies such that exactly 1 score per task (instead of 20) is included. This unreplicated data is often all you get from benchmarks.

Figure 2 displays the cumulative distribution function of the posterior mean of the evaluation bias for $n = 500$. The distribution is quite variant. There's a 46% chance that the posterior mean of β—the average increase in the log-odds of a correct prediction by pretraining on test set text instead of on independent text—is outside the interval $(-0.04, 0.04)$, which would indicate a significant negative or positive bias.[4] In other words, without subsampling, one may as well flip a coin to determine whether pretraining on unlabeled test set text is fair.

8 Conclusion

If the test set contains 200 observations, pretraining on text from the test set results in slight overoptimism compared to pretraining on independent text. This bias vanishes when the test set contains 500 observations. As a result, we recommended that each task in a few-shot learning benchmark includes a large set of unlabeled text which is used for pretraining, but not evaluation.

[4]For 0.04, the odds ratio is $e^{0.04} \approx 1.04$. For context, the average odds ratio between adjacent submissions in the RAFT leaderboard is 1.03. For posterior means outside $(-0.04, 0.04)$, all of their 89% credible intervals exclude 0, which evidences a non-null effect.

Another recommendation, which expands on the principle about robustness from Bragg et al. (2021), is based on the meta-analysis in §7: empirical studies of few-shot learning should consider including multiple, independent subsamples of training data. While a single training set combined with a large test set is sufficient for precise, unbiased estimation of out-of-sample performance, this estimator is conditional on the training set. In few-shot learning, the training set is, by definition, minimal. The estimator hides two sources of variance—that from the randomly drawn training set, and that from randomness inherent in the training procedure. Figure 2 shows that this variance is large-enough to turn a methodology into a coin flip for a standard pretraining-and-training procedure. Benchmarks should require training on multiple, independent subsamples to expose training variance.

An important limitation of this paper is that it does not analyze semi-supervised few-shot methods like Pattern-Exploiting Training. It also doesn't study more nefarious uses of the test set such as hand-inspecting the text and targeting interventions accordingly. This paper's conclusions are limited to task-adaptive pretraining of LMs before text classification with 100 labeled training examples.

A direction for future research is to vary the amount of labeled training examples. Perhaps there's more overoptimism for minimal training sets. Another direction is to explore the role of causality. Jin et al. (2021) argue and demonstrate that the benefit of task-adaptive pretraining depends on the causal direction of the learning task. Perhaps the principle of independent causal mechanisms is also relevant in assessing the fairness of pretraining on test set features.

Acknowledgements

Currently omitted for anonymity.

References

Neel Alex, Eli Lifland, Lewis Tunstall, Abhishek Thakur, Pegah Maham, C. Riedel, Emmie Hine, Carolyn Ashurst, Paul Sedille, Alexis Carlier, Michael Noetel, and Andreas Stuhlmüller. 2021. Raft: A real-world few-shot text classification benchmark. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300.

Jonathan Bragg, Arman Cohan, Kyle Lo, and Iz Beltagy. 2021. Flex: Unifying evaluation for few-shot nlp. *Advances in Neural Information Processing Systems*, 34:15787–15800.

Tomás Capretto, Camen Piho, Ravin Kumar, Jacob Westfall, Tal Yarkoni, and Osvaldo A Martin. 2022. Bambi: A simple interface for fitting bayesian linear models in python. *Journal of Statistical Software*, 103(15):1–29.

Emile Chapuis, Pierre Colombo, Matteo Manica, Matthieu Labeau, and Chloé Clavel. 2020. Hierarchical pre-training for sequence labelling in spoken dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2636–2648, Online. Association for Computational Linguistics.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. SemEval-2019 task 3: EmoContext contextual emotion detection in text. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 39–48, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. Climate-fever: A dataset for verification of real-world climate claims.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. 2023. MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4277–4302, Toronto, Canada. Association for Computational Linguistics.

Giovanni Grano, Andrea Di Sorbo, Francesco Mercaldo, Corrado A Visaggio, Gerardo Canfora, and Sebastiano Panichella. 2017. Android apps and user feedback: a dataset for software evolution and quality improvement. In *Proceedings of the 2nd ACM SIGSOFT international workshop on app market analytics*, pages 8–11.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022. PPT: Pre-trained prompt tuning for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8410–8423, Dublin, Ireland. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343, Brussels, Belgium. Association for Computational Linguistics.

Zhang Huangzhao. 2018. Yahoo-answers-topic-classification-dataset. `https://github.com/LC-John/Yahoo-Answers-Topic-Classification-Dataset`.

Zhijing Jin, Julius von Kügelgen, Jingwei Ni, Tejas Vaidhya, Ayush Kaushal, Mrinmaya Sachan, and Bernhard Schoelkopf. 2021. Causal direction of data collection matters: Implications of causal and anticausal learning for NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural*

Language Processing, pages 9499–9513, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. SemEval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 829–839, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Ravin Kumar, Colin Carroll, Ari Hartikainen, and Osvaldo Martin. 2019. Arviz a unified library for exploratory analysis of bayesian models in python. *Journal of Open Source Software*, 4(33):1143.

Nikola Ljubešić, Darja Fišer, and Tomaž Erjavec. 2019. The frenk datasets of socially unacceptable discourse in slovene and english.

P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65.

Irene Manotas, Ngoc Phuoc An Vo, and Vadim Sheinin. 2020. LiMiT: The literal motion in text dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 991–1000, Online. Association for Computational Linguistics.

Richard McElreath. 2018. *Statistical rethinking: A Bayesian course with examples in R and Stan*. Chapman and Hall/CRC.

Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. 2006. Spam filtering with naive bayes-which naive bayes? In *CEAS*, volume 17, pages 28–69. Mountain View, CA.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Richard D Morey, Rink Hoekstra, Jeffrey N Rouder, Michael D Lee, and Eric-Jan Wagenmakers. 2016. The fallacy of placing confidence in confidence intervals. *Psychonomic bulletin & review*, 23:103–123.

Amit Moscovich and Saharon Rosset. 2022. On the cross-validation bias due to unsupervised preprocessing. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(4):1474–1502.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

James O'Neill, Polina Rozenshtein, Ryuichi Kiryo, Motoko Kubota, and Danushka Bollegala. 2021. I wish I would have loved this one, but I didn't – a multilingual dataset for counterfactual detection in product review. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7092–7108, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. 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: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. 2006. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205.

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy. Association for Computational Linguistics.

Roshan Sharma. 2019. Twitter-sentiment-analysis. `https://github.com/sharmaroshan/Twitter-Sentiment-Analysis`.

Tan Thongtan and Tanasanee Phienthrakul. 2019. Sentiment classification using document embeddings trained with cosine similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 407–414, Florence, Italy. Association for Computational Linguistics.

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32, Prague, Czech Republic. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2021. Revisiting few-sample bert fine-tuning. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

A Classification tasks

The experiment was ran on 25 publicly available text classification tasks found in `https://huggingface.co/datasets`. Inclusion criteria:

1. All text is in English.
2. The number of classes is not greater than 25, because only 100 observations are used for training the classifier.
3. The task is to classify one text, not a pair as in, e.g., textual entailment tasks.
4. Texts aren't so long that too much useful signal is dropped when text is truncated to fit in BERT/GPT-2's context window, which is set to 256 tokens.
5. Based on our best judgment, it's likely that BERT/GPT-2 can do better than guessing, i.e., the task is not too niche.

Table 2 lists the exact tasks.

B Other experiment choices

This section expands on §4.

For $n = 500$, the number of epochs for pretraining was reduced from 2 (for $n = 200$) to 1. This choice was made to keep the number of pretraining steps (gradient updates) somewhat constant, so that the primary difference between $n = 200$ and $n = 500$ is the introduction of fresh data.

To clarify how classification training is performed: for BERT, the added linear layer transforms the `[CLS]` token embedding. For GPT-2, the added linear layer transforms the last token's embedding. The output dimension is the number of classes. All LM weights and the added linear layer's weights are finetuned.

`train` is stratify-sampled by the class to ensure every class is represented, and to reduce the variance of accuracy estimators. `test` is not stratify-sampled. We're only interested in the *difference* between accuracies, which is a function of the difference between model likelihoods because the priors are uniform. So even if accuracies are worse than the majority vote, differences are still meaningful for the purposes of this experiment.

`train` text is not included during pretraining to minimize the overlap of pretraining between $\text{acc}_{\text{extra}}$ and acc_{test}. This choice was made in an effort to widen any gap between them. The experiment tries to go out of its way to provide evidence of an effect.

`train` contains 100 observations. While this stretches the intention of "few" in few-shot learning, it allowed for much lower-variance comparisons based on initial experiments. BERT is quite sensitive—see Appendix D.2.

For $n = 500$, the number of subsamples was reduced from 50 (for $n = 200$) to 20 to reduce GPU time. Any potential increase in variance is presumably negligible: the number of subsamples decreased by a factor, but the test set size increased by the same factor. And the hierarchical model should expose much of the variance anyway.

C Hyperparameters and reproducibility

This paper's experiment and analysis code is available here: `https://github.com`.

`experiment.sh` lists hyperparameters used for each classification task and experiment configuration. Hyperparameters were pre-specified based on Zhang et al. (2021), and to obey memory limits. Run the script on a GPU with at least 15 GB VRAM to reproduce results in §5. It takes 50 hrs on a T4 GPU. Training is performed using the `transformers` package (Wolf et al., 2020).

The accuracy data analyzed in this paper is in

Hugging Face dataset	Author(s)	Number of classes	Text length (25, 75) percentiles
ag_news	Zhang et al. (2015)	4	(196, 266)
SetFit/amazon_counterfactual_en	O'Neill et al. (2021)	2	(60, 125)
app_reviews	Grano et al. (2017)	5	(10, 77)
blog_authorship_corpus	Schler et al. (2006)	2	(92, 556)
christinacdl/clickbait_notclickbait_dataset		2	(46, 69)
climate_fever	Diggelmann et al. (2020)	4	(80, 156)
aladar/craigslist_bargains	He et al. (2018)	6	(346, 713)
disaster_response_messages		3	(74, 178)
emo	Chatterjee et al. (2019)	4	(44, 83)
dair-ai/emotion	Saravia et al. (2018)	6	(53, 129)
SetFit/enron_spam	Metsis et al. (2006)	2	(342, 1553)
financial_phrasebank	Malo et al. (2014)	3	(79, 157)
classla/FRENK-hate-en	Ljubešić et al. (2019)	2	(34, 160)
hyperpartisan_news_detection	Kiesel et al. (2019)	2	(39, 63)
limit	Manotas et al. (2020)	2	(53, 123)
AmazonScience/massive	FitzGerald et al. (2023)	18	(24, 44)
movie_rationales	DeYoung et al. (2020)	2	(2721, 4659)
mteb/mtop_domain	Muennighoff et al. (2023)	11	(26, 44)
ccdv/patent-classification	Sharma et al. (2019)	9	(441, 775)
rotten_tomatoes	Pang and Lee (2005)	2	(76, 149)
silicone	Chapuis et al. (2020)	4	(29, 75)
trec	Wang et al. (2007)	6	(36, 61)
tweets_hate_speech_detection	Sharma (2019)	2	(62, 107)
yahoo_answers_topics	Huangzhao (2018)	10	(58, 213)
yelp_review_full	Zhang et al. (2015)	5	(287, 957)

Table 2: Brief descriptions of the 25 classification tasks used in this experiment. Click the link in the cell to be taken to the dataset homepage in https://huggingface.co/datasets. The dataset subset (or config) and the chosen prediction task are specified in code in src/pretrain_on_test/_load_data.py.

`analysis/accuracies_from_paper`.

To reproduce the analysis results in §6, run the following in sequence: `analysis/main_200.ipynb`, `analysis/main_500.ipynb`, `analysis/posterior_pred.ipynb`.

To reproduce the meta-analysis results in §7, run `analysis/meta.py` and then `analysis/meta.ipynb`.

D Results

First, a note re Figure 1: it's unexpected that the pretraining boost for $n = 200$ (top-right distribution) is larger than that for $n = 500$ (bottom-right distribution). It's reasonable to expect a larger pretraining boost for a larger pretraining set.

One explanation for this result is the reduction in pretraining epochs, as explained in the second paragraph of Appendix B. This choice was made to keep the number of pretraining gradient updates somewhat constant. Perhaps pretraining on $n = 500$ texts for 2 epochs increases the pretraining boost. It's unclear whether this change would also increase the evaluation bias for $n = 500$.

D.1 Individual analysis

The Jupyter notebook `analysis/dataset.ipynb` can be run to (1) produce visualizations of the distributions of $\text{acc}_{\text{extra}}$, acc_{test}, and acc_{base} (for each classification task and experiment configuration), and (2) compute p-values for the following hypothesis test:

$$H_0 : \text{E}[\text{acc}_{\text{test}} - \text{acc}_{\text{extra}}] = 0$$
$$H_1 : \text{E}[\text{acc}_{\text{test}} - \text{acc}_{\text{extra}}] > 0.$$

The p-value is estimated via permutation testing. It's then adjusted to control the false discovery rate (Benjamini and Hochberg, 1995). No p-values were statistically significant at the 0.05 level.

Care has to be taken when attempting to analyze or interpret $\text{acc}_{\text{extra}}$ - acc_{base} and acc_{test} - $\text{acc}_{\text{extra}}$ *together*. That's because these differences are not independent:[5] if $\text{acc}_{\text{extra}}$ is high, then $\text{acc}_{\text{extra}}$ - acc_{base} increases and acc_{test} - $\text{acc}_{\text{extra}}$ decreases. This paper does not analyze the scores together, per se. We care about acc_{test} - $\text{acc}_{\text{extra}}$. $\text{acc}_{\text{extra}}$ - acc_{base} only exists to sanity check that the pretraining code works; there is an effect to detect.

[5] We realized this after plotting the two differences on a scatter plot, and being wrongly surprised by the clear correlation :-)

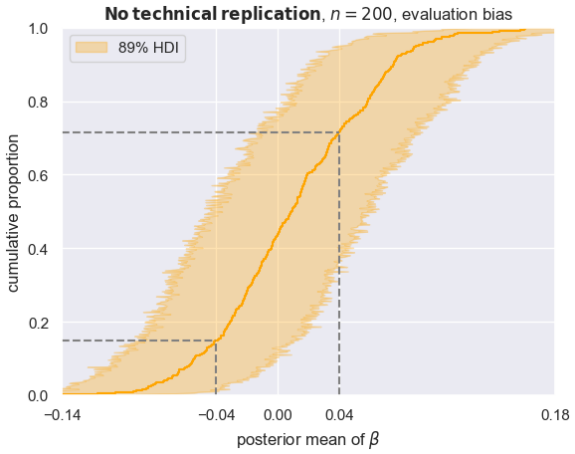


Figure 3: evaluation bias is akin to acc_{test} - $\text{acc}_{\text{extra}}$.

D.2 Difference distributions

Figure 7 visualizes the distributions of the paired differences—$\text{acc}_{\text{extra}}$ - acc_{base} and acc_{test} - $\text{acc}_{\text{extra}}$–for each classification task and LM type for $n = 200$. Figure 8 does the same for $n = 500$.

E Hierarchical model checks

Hierarchical models require some basic checks to have faith in their results (McElreath, 2018).

The posterior samples of β which were used (in part) to draw posterior predictive samples—whose differences, $\bar{\hat{Y}}_{\dots 1} - \bar{\hat{Y}}_{\dots 0}$ (dot notation), are plotted in Figure 1—are shown in Figure 6. All trace plots are healthy.

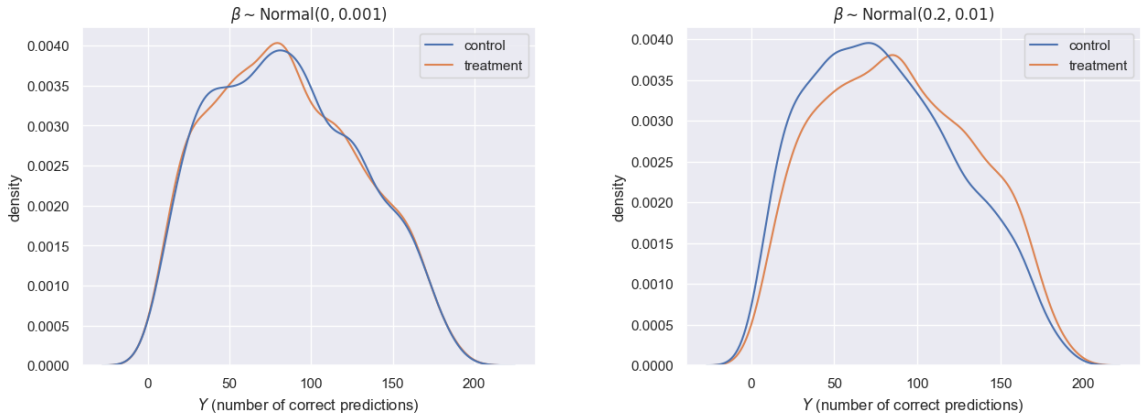
Figure 4 contains prior predictive distributions for $n = 200$, demonstrating that priors are not unreasonable. Using default priors from the `bambi` package (Capretto et al., 2022), while scientifically unreasonable (because they result in wide, basin-like accuracy distributions), did not change the conclusions of this paper.

Figure 5 contains posterior distributions of β for $n = 200$, demonstrating the hierarchical model's ability to recover both null and non-null effects.

To reproduce Figure 4 and Figure 5, run the Jupyter notebook `analysis/test.ipynb`.

F Meta-analysis

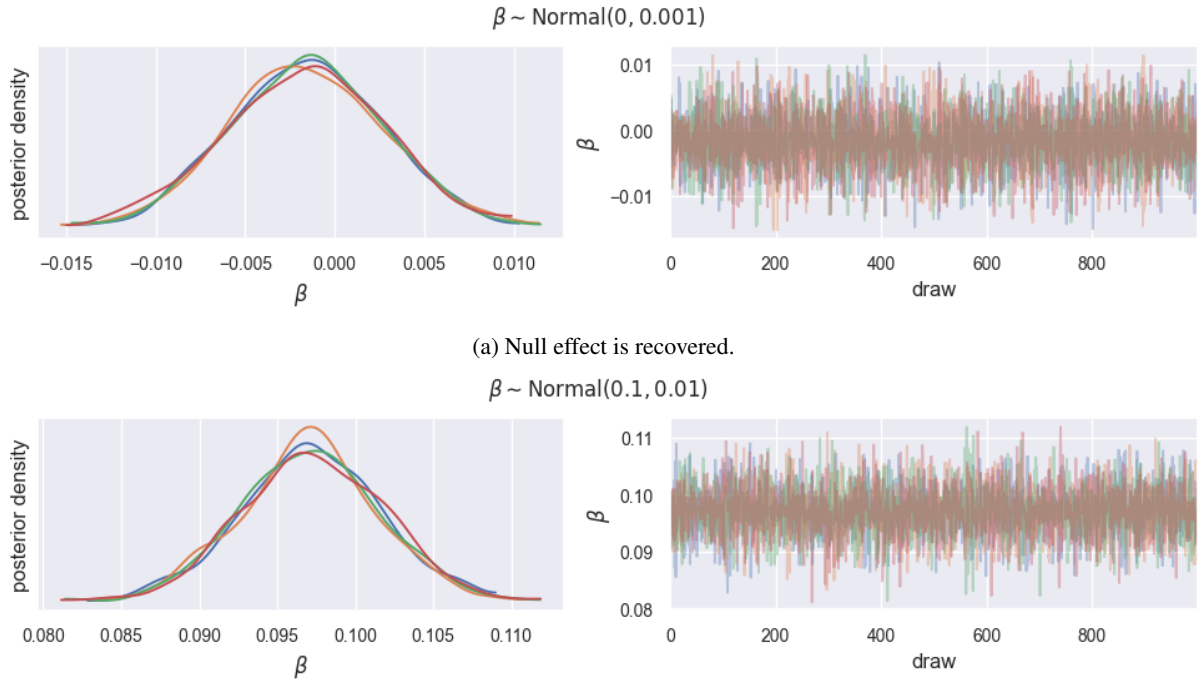
No divergences were observed. Figure 3 is analogous to Figure 2. There's slightly more variance in the posterior mean of β than in $n = 500$ because the size of the test set decreased. For $n = 200$, 43% of means were outside $(-0.04, 0.04)$. For these means, 81% of their 89% credible intervals exclude 0.



(a) Null effect

(b) Non-null, positive effect

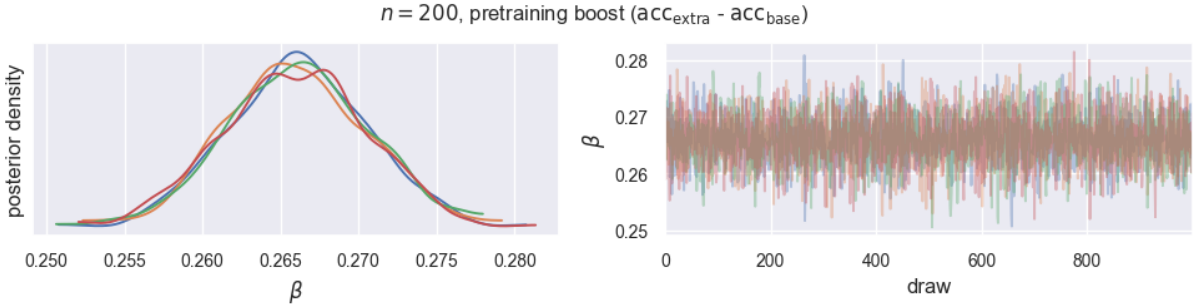
Figure 4: Prior predictive distributions for $n = 200$ from two different priors for β—the expected increase in the log-odds of a correct prediction resulting from an intervention/treatment.



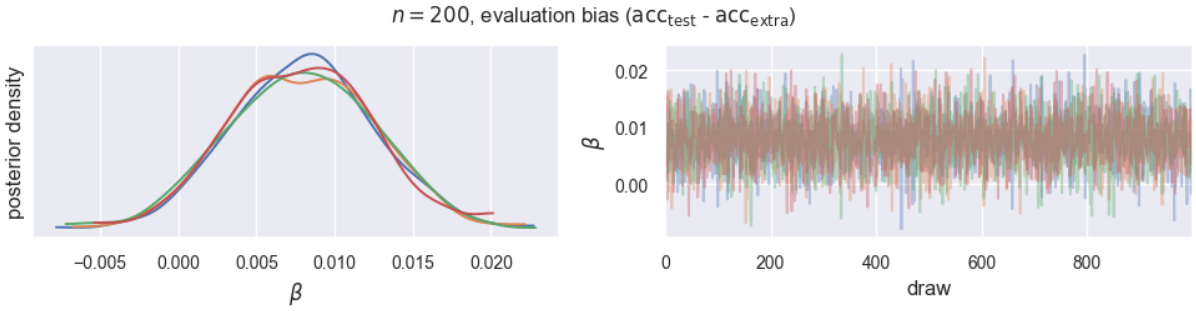
(a) Null effect is recovered.

(b) Non-null, positive effect is recovered.

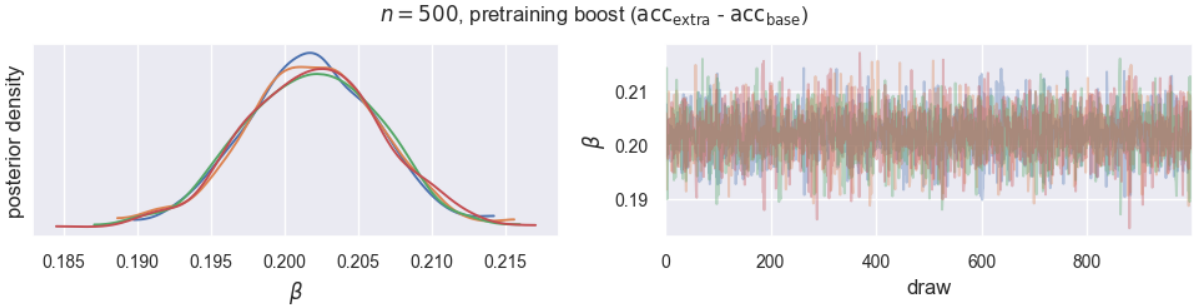
Figure 5: Posterior distributions and trace plots for null and non-null effects **from simulated data** where $n = 200$, approximated by four chains with 1,000 draws each, after 500 steps of tuning. For each model, no divergences were observed during the fitting procedure. Visualizations were produced by the `arviz` package (Kumar et al., 2019).



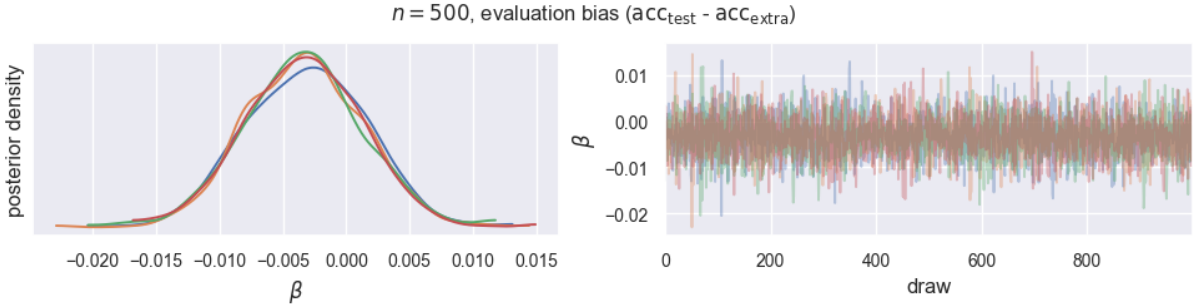
(a)



(b)



(c)



(d)

Figure 6: Posterior distributions and trace plots for the effects of interest, approximated by four chains with 1,000 draws each, after 500 steps of tuning. For each model, no divergences were observed during the fitting procedure.

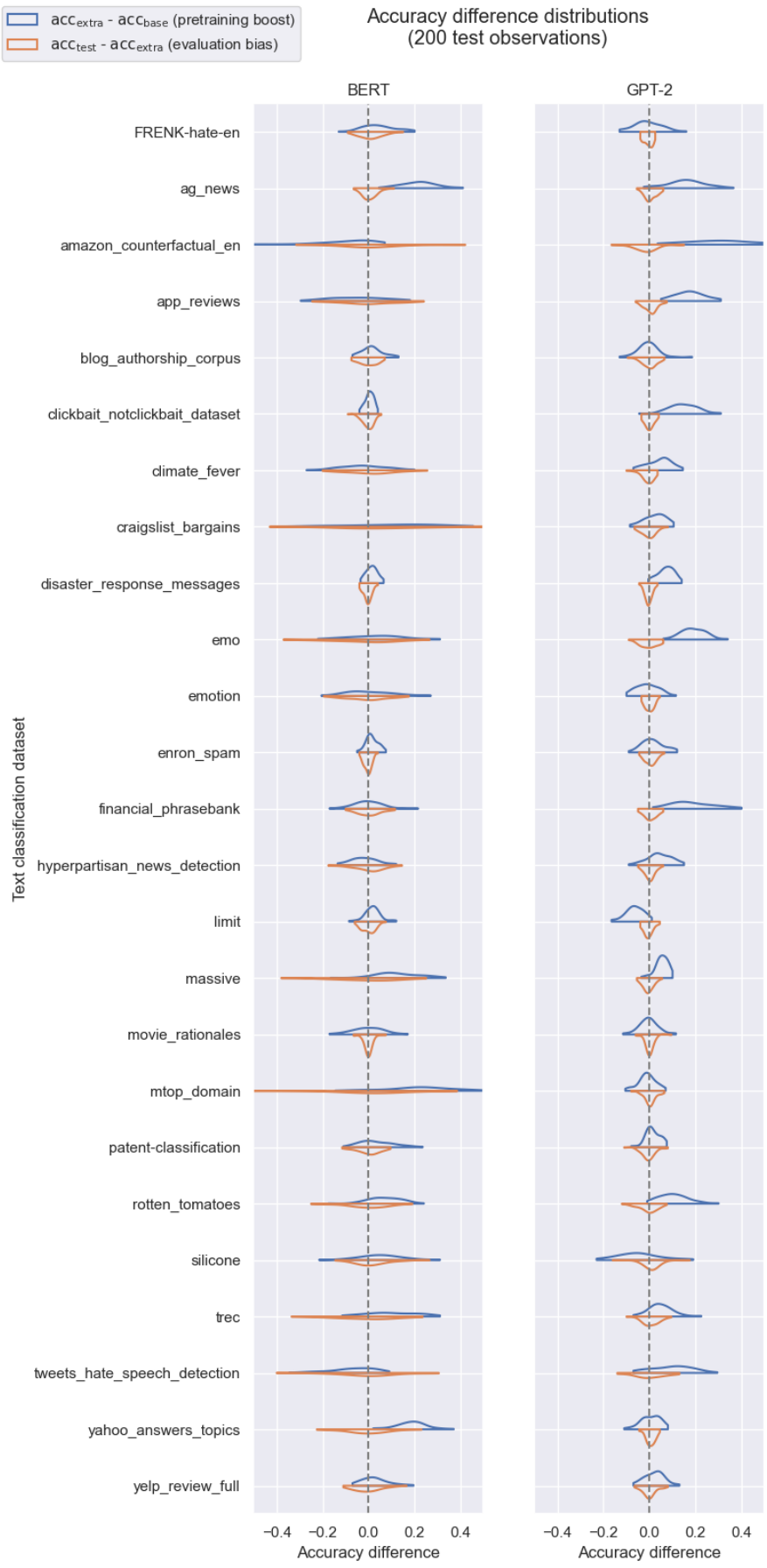

acc_extra - acc_base (pretraining boost)
acc_test - acc_extra (evaluation bias)
Accuracy difference distributions
(200 test observations)
BERT
GPT-2
FRENK-hate-en
ag_news
amazon_counterfactual_en
app_reviews
blog_authorship_corpus
clickbait_notclickbait_dataset
climate_fever
craigslist_bargains
disaster_response_messages
emo
emotion
enron_spam
financial_phrasebank
hyperpartisan_news_detection
limit
massive
movie_rationales
mtop_domain
patent-classification
rotten_tomatoes
silicone
trec
tweets_hate_speech_detection
yahoo_answers_topics
yelp_review_full
Text classification dataset
−0.4 −0.2 0.0 0.2 0.4
Accuracy difference
−0.4 −0.2 0.0 0.2 0.4
Accuracy difference

Figure 7

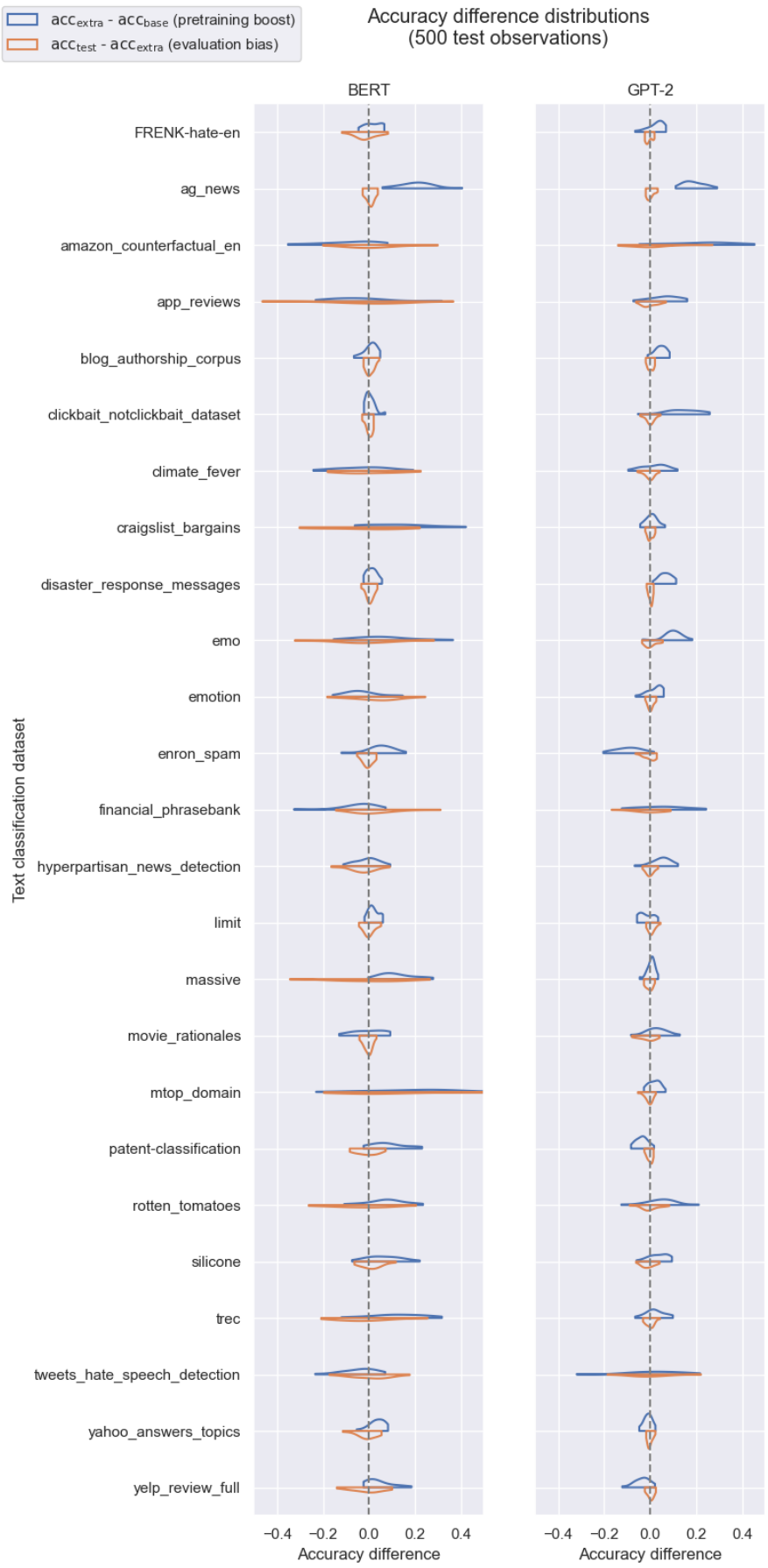

acc_extra - acc_base (pretraining boost)
acc_test - acc_extra (evaluation bias)
Accuracy difference distributions
(500 test observations)
BERT
GPT-2
FRENK-hate-en
ag_news
amazon_counterfactual_en
app_reviews
blog_authorship_corpus
clickbait_notclickbait_dataset
climate_fever
craigslist_bargains
disaster_response_messages
emo
emotion
enron_spam
financial_phrasebank
hyperpartisan_news_detection
limit
massive
movie_rationales
mtop_domain
patent-classification
rotten_tomatoes
silicone
trec
tweets_hate_speech_detection
yahoo_answers_topics
yelp_review_full
Text classification dataset
−0.4 −0.2 0.0 0.2 0.4
Accuracy difference
−0.4 −0.2 0.0 0.2 0.4
Accuracy difference

Figure 8