On Robustness-Accuracy Characterization of Large Language Models using Synthetic Datasets

Ching-Yun Ko [1] Pin-Yu Chen [2] Payel Das [2] Yung-Sung Chuang [1] Luca Daniel [1]

Abstract

Despite the impressive capability of large language models (LLMs) in solving different downstream tasks, new concerns about proper performance evaluation have been raised, especially for test-data leakage caused by accidentally including them during pretraining, or by indirectly exposing them through API calls for evaluation. Motivated by these, in this paper, we propose a new evaluation workflow that generates steerable synthetic language datasets and proxy tasks for benchmarking the performance of pertained LLMs on sentence classification tasks. This approach allows for better characterization of the joint analysis on the robustness and accuracy of LLMs without risking sensitive information leakage. Verified on various pretrained LLMs, the proposed approach demonstrates promising high correlation with real downstream performance.

1. Introduction

While new opportunities present themselves with foundation models (not only LLMs), they also bring forth potential risks and challenges (Bommasani et al., 2021; Blodgett & Madaio, 2021; Wiggins & Tejani, 2022; Thieme et al., 2023; Biderman et al., 2023). Specifically, Dr. Percy Liang, a prominent researcher and Director of the Center for Research on Foundation Models at Stanford University, recently took to Twitter to express his concerns about the potential for language models to be trained on test sets. Furthermore, even private or held-out unpublished test sets may be vulnerable to data leakage through querying the LLMs via APIs for evaluation purposes. Recently, the generative embedding inversion attack (Li et al., 2023) that reconstructs input sequences based on sentence embeddings further deepened our concerns about information leakage during test time.

To address this caveat of "information leakage" leading to improper and fragile evaluation, in this paper, we propose a synthetic testbed for benchmarking two critical aspects of LLM sentence embeddings: accuracy and robustness. We further propose a novel approach to generate synthetic datasets for LLMs that can serve as proxy test sets. Our approach leverages existing sentiment lexicons, such as SentiWordNet 3.0 (Baccianella et al., 2010), to generate working word lists based on the word (or synset) level labels. We build positive, negative, and neutral word lists from SentiWordNet 3.0, and use them to design synthetic datasets for LLM evaluation. Our synthetic dataset generation follows the nesting parentheses (Papadimitriou & Jurafsky, 2020), which mimics the recursion structural hypothesis about the narrow language faculty in humans (Hauser et al., 2002) and the dependency tree structure in natural language (Chiang & Lee, 2022). By maneuvering the mixing percentage of binary words (positive/negative words) and neutral words, we create a configurable testbed for evaluating the performance of LLMs on different levels of difficulty and complexity. Specifically, we benchmark and quantify the ability of each LLM on sentence classification tasks by comparing their performance on a set of our synthetic datasets with varying difficulty levels. It is worthwhile to note that since this work focuses on benchmarking LLMs on sentence classification tasks, our synthetic datasets admittedly do not try to encode syntax. This relaxation is inspired by our experiment where we noticed that 86% of the labels given by Huggingface sentiment analysis pipeline on product reviews classification (CR) (Hu & Liu, 2004) remain the same after removing 284 stop words (listed in the supplementary materials) from the sentences. We dub our framework of benchmarking LLMs using synthetic texts by *SynTextBench* and present the workflow in Figure 1. With the popularity of LLMs, we argue that they should be subject to much more rigorous and comprehensive testing and auditing before being deployed in real-life applications (Weidinger et al., 2021; Ganguli et al., 2022; Mökander et al., 2023; Rastogi et al., 2023). The evaluation framework using synthetic data outlined in this paper should be viewed as a contribution towards ensuring independent and more sustainable LLM auditing.

Our **main contributions** are:

- We introduce *SynTextBench*, a novel theoretically-grounded framework to generate steerable synthetic datasets towards a holistic evaluation of LLMs. The use of synthetic datasets alleviates the risk of test-data leakage and offers

[1]MIT [2]IBM Research. Correspondence to: Ching-Yun Ko <cyko@mit.edu>.

Work presented at the ES-FoMo Workshop at ICML 2023.

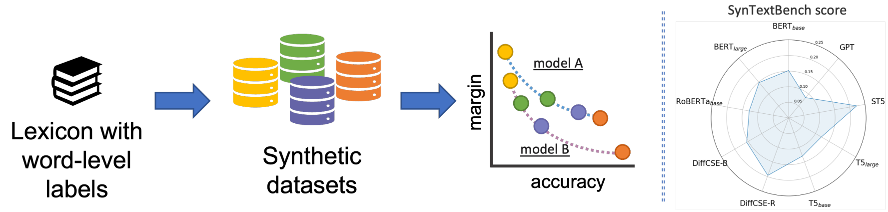


Figure 1: Overview of SynTextBench. SynTextBench generates a set of synthetic datasets from any given lexicon with word-level labels. We test the given LLM on these datasets and obtain robustness-accuracy characterization under a range of steerable task difficulties. For each LLM, we can plot the robustness-accuracy trade-off curve and make model comparisons.

new tools for LLM testing and auditing.

- SynTextBench provides a configurable lightweight testbed and a quantifiable metric for evaluating the robustness and accuracy of LLMs on different levels of difficulty and complexity for sentence classification tasks, with no restrictions on the model architecture.
- We conduct experiments with several state-of-the-art LLMs on our testbed and report their performance and behavior. SynTextBench, as a real-data-free evaluation method, shows high correlation with robustness-accuracy performance evaluated on real data. Further study demonstrates its capability of making quick attribution comparisons such as analyzing fine-tuning effects for LLMs.

2. Methodology

We defer the review of sentence representations, pretrained model evaluations, sentiment lexicons, and robust Bayes optimal classifiers to the appendix.

2.1. Why using synthetic datasets for LLM evaluation?

To reduce the reliance on real-life data, we build synthetic tasks by generating synthetic sentences as model inputs at test time. This way, we no longer need to exchange private or label-annotated data as test sets with LLM APIs. We detail the desiderata of proxy tasks and the evaluation metric.

- Task substance: Tasks should test a pretrained LLM's ability to encode sentence representations that preserve class separability when evaluated by a linear classifier.
- Task difficulty: Tasks' difficulty should be configurable to allow for comprehensive analysis, i.e., one can generate tasks of various levels of difficulty.
- Task feasibility: Tasks should be feasible to solve, i.e., the sentences should be distinguishable to a certain degree by an algorithm that works on the raw sentences input.
- Task independence: Tasks' ground-truth should be independent of the LLM to be evaluated, in order to avoid biased evaluation, e.g., the label in the task should not be given by an LLM.
- Task equity: Tasks should be able to be generated by anyone and affordable for anyone without requiring any private data or favoring any party with more resources.
- Metric informativeness: The designed framework should give a metric with a clear implication (the larger the better) and correlates well with the real performance.

With these in mind, it is straightforward to see why we should not opt for synthetic datasets generated by any LLM: (1) task difficulty would not be configurable, (2) the evaluation might favor the LLM that generates the synthetic sentences and/or the pseudo-labels (causing label leakage), and (3) any auditor without access to proprietary LLMs or datasets cannot run independent evaluation.

2.2. Constructing synthetic datasets and tasks

Word List. Building a synthetic task requires us to define the synthetic input data to be used. Here, we utilize sentiment lexicons with word-level labeling such as SentiWordNet 3.0 (Baccianella et al., 2010) to build positive, negative, and neutral word lists. We give details and running examples in the appendix. To this end, we created the word lists from SentiWordNet 3.0 as depicted in Figure 2(a). Next, we explain the structure of the synthetic sentences.

Sentence structure. Motivated by a recent literature (Papadimitriou & Jurafsky, 2020) that explored the power of non-linguistic artificial parentheses languages in training models that transfer to NLP tasks, we follow nesting parenthesis when generating the synthetic sentences in our proxy tasks. Specifically, nesting parenthesis involves paired tokens and a recursive structure. For example, by referring to Figure 2(b), one sees that t_1 and t_4 are paired words, while t_2 and t_3 are another paired words. In our example, words are hierarchically nested, meaning the token to be paired with t_2, which is t_3 herein, should appear before the pairing token with t_1. In other words, it observes a "last in first out" data structure, and the arcs in Figure 2(b) do not cross.

Sentence generation and difficulty level. Now we explain how to do sentence generation following the structure. Let us revisit the case in Figure 2(b). Assume we already generated the first five tokens $t_1 : t_5$ in a positive sentence ($y = 1$) with colors denoting the picked word. To decide the next token, we sample t_6 from a mixing distribution D, where $D = p_e \cdot$ '<eos>' $+ p_n(1-p_e) \cdot \text{last\_unpaired\_word} + (1-p_n)(1-p_e) \cdot D_{\text{new}}$. To interpret distribution D, we see there are 3 possible outcomes for the incoming t_6 token: (1) the end of sentence indicator '<eos>', (2) the popped token from the stack that stores the unpaired words, i.e., the last unpaired word, (3) a new word. If it is to pick a new word, this word will be sampled from the distribution of new words D_{new}, which directly depends on the label y of the sentence to be generated and the desired task difficulty. For a positive sentence ($y = 1$), $D_{\text{new}|y=1}$ is described by the

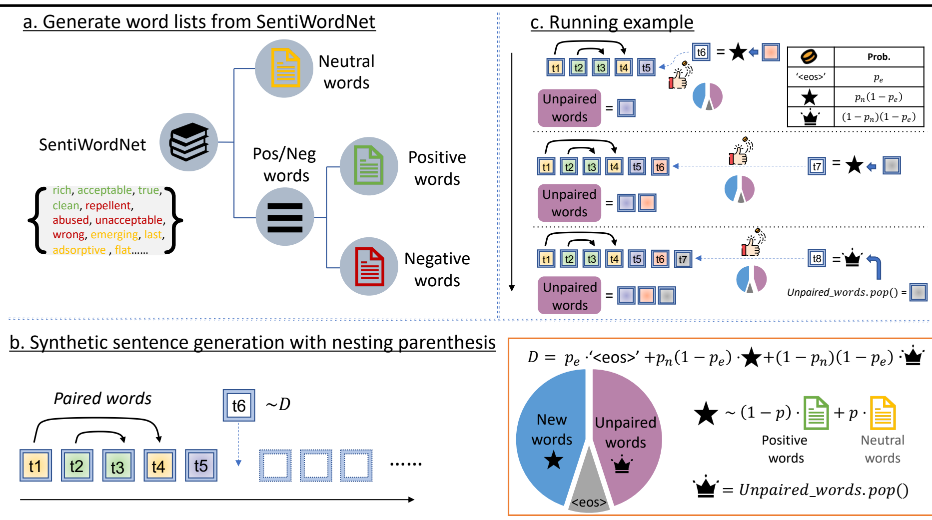


Figure 2: Overview of the sentence generation procedure. In block a, we generate word lists from SentiWordNet 3.0. In block b, we generate each sentence token following nesting parenthesis and mixing distribution D. In block c, we show a running example of sequentially generating t_6, t_7, t_8.

PDF $p \cdot f_{\text{NEU}}(x) + (1-p) \cdot f_{\text{POS}}(x)$, where p specifies the percentage of neutral words in a synthetic sentence, f_{NEU} and f_{POS} give the PDFs of neutral and positive words. Similarly, if we are to generate a negative sentence ($y = -1$), we have $D_{\text{new}|y=-1}$ described by $p \cdot f_{\text{NEU}}(x) + (1-p) \cdot f_{\text{NEG}}(x)$, where f_{NEG} gives the PDF of negative words.

In essence, the label of the generated sentence is determined by construction, which guarantees the **task independence**. It also allows configurable **task difficulty** by adjusting the percentage p of neutral words in a synthetic sentence. That is, it is easier to predict the sentiment of sentences consisting of 90% positive words and 10% neutral words than that of sentences constructed all by neutral words. We refer readers to the appendix for exemplary synthetic examples. In Figure 2(c), we show a running example of the sentence generation process, where we flip a coin with 3 outcomes each time to decide on a new token. When the realization is "new words" (like in t_6 and t_7), this word will also be pushed to the stack "*Unpaired_words*" that stores unpaired words. When we are deciding t_8, we draw "unpaired words" and hence t_8 is determined by *Unpaired_words.pop()*. In the appendix Figure 4, we prove the **task feasibility** by demonstrating the separability of generated synthetic datasets by SentiWordNet sentiment analysis algorithm (Denecke, 2008). With an increasing mixing ratio p, while the task becomes harder, we show there at least exists an algorithm that can separate the data to a certain degree, showcasing a lower bound on the optimal classification strategy. By our workflow of constructing synthetic datasets and tasks, we also guarantee **task equity** since the generation process requires no access to any LLM or private data, and can be replicated by anyone with limited resources. Furthermore, we note that the construction of synthetic datasets and tasks described herein is also extendable to other values by swapping the lexicon used for extracting word lists.

2.3. Robustness-accuracy evaluation

Given an LLM g, let x, y be the input sentence and its label, z be the sentence embeddings $z = g(x) \in \mathbb{R}^n$, we are interested in evaluating the accuracy of the sentence embedding classifiers f, and the average distance Δ from sentence embeddings to the classifiers. We let z_1 be $\{z : z = g(x), y = 1\}$ and z_{-1} be $\{z : z = g(x), y = -1\}$.

Preparing sentence embeddings. Recall that Bert-flow (Li et al., 2020) and Bert-whitening (Su et al., 2021) transformed the sentence embeddings into an isotropic Gaussian distribution to remedy the anisotropic behavior in the sentence embedding vector space. We thereby also perform whitening on sentence representations before we draw the decision rule on the embeddings. Transforming a set of sentence embeddings of a class into an isotropic Gaussian involves two steps: (1) model the mean b_y and covariance Σ_y of original embeddings z_y, (2) apply a transformation to the embeddings $F^T S^{-1/2} z_y$, where $FSF^T = \Sigma_y$ is the singular value decomposition of Σ_y. Since Σ_y can be ill-conditioned, directly applying $S^{-1/2}$ on embeddings z_y might amplify noisy signals due to numerical instability. Thus, we propose to reduce the dimension according to energy-preservation (Leskovec et al., 2020). We select to keep K dimensions according to $\arg\min_k \frac{\sum_{i=1}^k s_i}{\sum_{i=1}^n s_i} \geq 0.99$, where $s_i = \text{diag}(S)[i]$ is the i-th largest singular value of S. Till now, we see that the sentence embeddings are transformed to a $\mathbb{R}^K$ vector space via $F^T_{:,1:k} S^{-1/2}_{1:k,1:k} z_y$. We perform these operations for both classes ($y = 1$ and $y = -1$) separately. Since we want the transformed embeddings to observe the original relative distance between two classes, we further scale the distance between two whitened Gaussians by $d_{\underline{\text{Inter}}\text{-class}}/d_{\underline{\text{Intra}}\text{-class}}$, where the numerator $d_{\underline{\text{Inter}}\text{-class}} = \|b_1 - b_{-1}\|$ calculates the inter-class distance (the distance between two class centers b_1 and b_{-1}), and the denominator $d_{\underline{\text{Intra}}\text{-class}} = \frac{1}{m_1+m_2}(\sum_{i=1}^{m_1} \|z_1^i - b_1\| + \sum_{j=1}^{m_2} \|z_{-1}^j - b_{-1}\|)$ calculates the intra-class distance (the average distance from class data to class mean) with m_1 and m_2 being the number of positive sentences and negative sentence, respectively. We let T_y denote the overall transformation operations and obtain transformed embeddings $\hat{z_1} = T_1(z_1)$ and $\hat{z_{-1}} = T_{-1}(z_{-1})$.

Decision margins induced by robust Bayes optimal classifiers. Recall that robust Bayes optimal classifiers explicitly give the optimal classification strategy for class-conditional Gaussian distribution in the presence of data perturbations (Bhagoji et al., 2019; Dan et al., 2020). Here, we see that $(\hat{z}, y)$ are modeled as P_{μ_1,μ_2,I_K}: $\hat{z}|y = 1 \sim \mathcal{N}(\mu_1, I_K), \hat{z}|y = -1 \sim \mathcal{N}(\mu_2, I_K)$, and $y \in \mathcal{C} = \{+1, -1\}$. While finding the robust Bayes optimal classifier generally involves solving an optimization problem, when the covariance is an identity matrix, the class priors $\mathbb{P}(y = 1) = \tau$, $\mathbb{P}(y = -1) = 1 - \tau$, and the perturbation radius ϵ, the optimal classifier is given as $\text{sign}(w^T(\hat{z} - \frac{\mu_1+\mu_2}{2}) - q/2)$, where $q = \log\{(1-\tau)/\tau\}$, $w = \tilde{\mu}(1 - \epsilon/\|\tilde{\mu}\|_2)$, and $\tilde{\mu} = \frac{\mu_1-\mu_2}{2}$. Furthermore, when the classes are balanced, the (robust) Bayes optimal classifier is $\text{sign}(\tilde{\mu}^T(\hat{z} - \frac{\mu_1+\mu_2}{2}))$, which is independent of ϵ. We use this classifier to calculate the accuracy on synthetic datasets. In fact, we prove in the appendix that ϵ-robust Bayes optimal classifiers overlap for all ϵ as long as $\tilde{\mu}$ lies completely within a degenerate subspace of the eigenspace of the covariance matrix. In the case of an identity covariance matrix, the degenerated subspace of the eigenspace expands the whole $\mathbb{R}^K$, hence $\tilde{\mu}$ lies in the space naturally.

Now that we have specified the optimal robust classification rule on the transformed sentence embeddings, we write out the decision margin induced by the classifiers using an informal but more intuitive statement: for any sample z, the Bayes optimal classifier f of class-balanced class-conditional Gaussian distribution P_{μ_1,μ_2,I_K}, yields a decision margin of $\|\Delta\|_2 = \frac{|(\hat{z}-\frac{\mu_1+\mu_2}{2})^T\tilde{\mu}|}{\|\tilde{\mu}\|_2}$, and if we scale the margin by the distance between two Gaussian centers, we obtain a scaled margin of $\|\bar{\Delta}_z\|_2 = \frac{|(\hat{z}-\frac{\mu_1+\mu_2}{2})^T\tilde{\mu}|}{\|\tilde{\mu}\|_2^2}$. We give the formal results for the generic class prior in the appendix. In the following, we will state the complete algorithm for characterizing robustness-accuracy performance (cf. Section 2.3) of LLMs using synthetic datasets (cf. Section 2.2).

2.4. SynTextBench score and algorithm

In our benchmarking process, we essentially generate a sequel of tasks with different difficulty levels and inspect how the magnitude of decision margins changes with the classifier accuracy. In terms of robustness-accuracy characterization, it is desirable for an LLM to consistently yield high classification accuracy, while maintaining a big decision margin (that is, less sensitive to perturbations in the embedding space). The pseudocode of the proposed framework, *SynTextBench*, is given in Algorithm 1. In practice, we let $P = \{0, 0.05, \ldots, 0.9, 0.95\}$, and subsequently generate 20 synthetic datasets with $p = 0$ being the easiest and $p = 0.95$ being the hardest. Then, we perform analysis on the sentence embeddings of various synthetic datasets, and threshold the accuracy at a_T based on utility. The threshold serves as a penalty for poor sentence embeddings that lead to an undesirable accuracy under this threshold, matching our **task substance** of testing LLM's ability to preserve linear separability. By referring to Figure 1, Line 1 in Algorithm 1 determines the word lists from a given lexicon. From Line 2 to Line 9, the for-loop generates one synthetic dataset at one time, on which we compute an (accuracy, avg. margin) pair (a_p, δ_p) and draw one point on the margin-accuracy 2D plot as in Figure 1. Since we not only care about the curvature of the curve but also how the (accuracy, avg. margin) pairs span on the curve, we define a goodness function $s(a) = \frac{1}{|P|}\sum_{\{p \in P, a_p > a\}} \delta_p$ on $\mathbb{R}[0, 1]$ to account for the span. By our definition, $s(a)$ will be a monotonically decreasing function (e.g., Figure 6) and calculate the expected margin conditioned on the accuracy level. The final SynTextBench score is defined by the integration over the desirable range of threshold accuracy, i.e. SynTextBench score $= \int_{a_T}^{1} s(a)da$. We use SynTextBench as a quantifiable score to inform the accuracy-robustness aspect of a pretrained LLM. We apply Algorithm 1 on various models to evaluate their performance. In the later section, we will demonstrate the **metric informativeness** by measuring the correlation between SynTextBench scores and the average real-life sentence classification task performance.

3. Experiments

In this section, we test SynTextBench on multiple pretrained LLMs to demonstrate the usage of the framework, including $\text{BERT}_{\text{base}}$, $\text{BERT}_{\text{large}}$, $\text{RoBERTa}_{\text{base}}$, DiffCSE-B, DiffCSE-R, T5_{base}, T5_{large}, ST5, and GPT. We give more details of these models in the appendix. For models that have an encoder component (encoder-only or encoder-decoder), we use the average output from the first and the last layer as sentence embeddings. For the decoder-only model, we use the embedding of the last token as sentence embeddings.

Baselines. We followed the implementation of (Whitney et al., 2020) and fed the pretrained LLMs with synthetic texts generated according to Section 2.2 and reported the validation accuracy (Val loss), minimum description length (MDL), surplus description length (SDL), and ϵ-sample complexity (ϵSC) as baselines (Blier & Ollivier, 2018; Voita & Titov, 2020; Whitney et al., 2020). Since these methods take one dataset as inputs, we choose a relatively easy synthetic proxy task generated by $p = 0.2$ as the input dataset.

Correlation with the real-life task performance. In order to demonstrate the power of SynTextBench in informing the robustness-accuracy performance of a given LLM on possible downstream sentence-level tasks, we test LLMs

Table 1: Correlation of real-data-free evaluation metric and real-data accuracy at different synthetic dataset sizes.

n	4096	8192	16384	32768
Val loss	0.29±0.50	0.65±0.00	0.61±0.01	0.27±0.02
MDL	0.57±0.11	0.52±0.04	0.51±0.03	0.48±0.03
SDL, ε=1	0.57±0.11	0.51±0.04	0.43±0.02	0.31±0.01
εSC, ε=1	-	-	-	-0.04±0.000
SynTextBench	**0.94±0.01**	**0.96±0.01**	**0.96±0.00**	**0.93±0.00**

Table 2: Aggregated correlation with real-data-free evaluation metrics and the aggregated robustness-accuracy performance, and its breakdown.

Correlation. w/	Rob.-Acc.	Rob.-STS	Rob.-Transfer
Val loss	-0.06±0.15	0.08±0.13	-0.13±0.24
MDL	0.64±0.06	0.55±0.08	0.62±0.03
SDL, ε=1	0.60±0.02	0.51±0.04	0.58±0.028
εSC, ε=1	-	-	-
SynTextBench	**0.76±0.04**	**0.76±0.03**	**0.69±0.05**

Table 3: Performance evaluation of T5 and ST5 by real-data-free metric (SynTextBench) and real-data-dependent metrics (accuracy and robustness on SentEval).

	SynTextBench				Real-life	
n	4096	8192	16384	32768	accuracy	robustness
T5	0.111±0.002	0.130±0.001	0.145±0.000	0.158±0.001	82.78	12.21
ST5	0.214±0.000	0.223±0.001	0.227±0.001	0.230±0.000	90.17	13.23

on SynTextBench as well as real-life tasks. Concretely, we applied Algorithm 1 and obtained one goodness function $s(a)$ for each LLM (Figure 6), from which the final SynTextBench score can be determined by definition. We refer readers to Table 6 in the appendix for the complete results. To gauge the performance of these pretrained LLMs on downstream real-life tasks, we evaluate given models on SentEval (Conneau & Kiela, 2018)) and show the detailed numbers in Figure 7 in the appendix. SentEval tasks include seven semantic textual similarity tasks (denoted by "STS tasks"), where results are given by the Spearman's correlation with output range $[-1, 1]$, and seven transfer learning tasks (denoted by "Transfer task"), where results are given by the standard accuracy with range $[0, 1]$. We scale the former to the same range as the latter, $[0, 1]$, and take an average as the final accuracy indicator. We put the full list of tasks in the appendix Table 5.

To demonstrate the utility of SynTextBench score, we list the Pearson correlation coefficients between real-data-free evaluation methods and the accuracy of SentEval tasks in Table 1. Five real-data-free metrics are considered that includes Val loss, MDL, SDL, εSC, and SynTextBench. Since the smaller the baselines are, the better, we add a negative sign when calculating their Pearson correlation coefficients. As we have the flexibility of generating synthetic datasets with various sizes (number of sentences), we compare four configurations $n = \{4096, 8192, 16384, 32768\}$. According to Table 1, SynTextBench consistently gives scores highly correlated with real-life task accuracy, with correlation coefficients that are above 0.9. For baselines, the highest correlation is when $n = 8192$ and evaluated by Val loss, 0.65. It is noteworthy that SynTextBench is a stabler metric as substantiated by the smaller standard deviation.

Furthermore, to evaluate LLMs' robustness performance, we use PWWS attack (Ren et al., 2019) and report the average percentage of perturbed words as the robustness indicator. Essentially, the attacker perturbs the inputs gradually by changing more and more words until the perturbation leads to a wrong classification result. We analyze the correlation on Transfer tasks when $n = 8192$ since these tasks are classification tasks where adversarial attacks are well-defined. To combine robustness correlation with accuracy correlation, we add up two ranking vectors by robustness and accuracy measures, and calculate its Pearson correlation with the ranking by one of the real-data-free evaluation metrics (Val loss, MDL, SDL, ϵSC, SynTextBench). This way, we effectively obtain the aggregated Spearman correlation coefficient between real-data-free evaluation metrics and joint robustness-accuracy performance. We refer readers to the appendix for more details. We list the results in Table 2. From the "Rob.-Acc." column, we see SynTextBench has an overall higher correlation with robustness-accuracy performance compared to other baselines. Recall that accuracy results were aggregated from STS and Transfer tasks. In Table 2, we show how each component contributes to the correlation. In the "Rob.-STS" and "Rob.-Transfer" columns, we use only STS or Transfer task results as the accuracy measure when ranking the models, and the remaining steps follow. From the two columns, we see that SynTextBench shows a stronger correlation compared to baselines, while having a better correlation with Robustness-STS accuracy performance than Robustness-Transfer accuracy performance.

Attribute comparisons. Besides having high correlation with real-life task performance, SynTextBench can be used to make model attribute comparisons. Table 3 lists the SynTextBench scores of pretrained T5 and ST5 under different dataset sizes n, together with the accuracy and robustness on SentEval tasks. From the table, it can be seen that the SynTextBench score of ST5 is significantly higher than that of T5 across all n, indicating contrastive fine-tuning is beneficial for improving sentence embeddings. This conclusion is in sync with the observations from real-life tasks, where we see ST5 yields both higher accuracy and robustness.

4. Conclusion

In this paper, we have proposed SynTextBench, a configurable real-date-free lightweight testbed for evaluating the accuracy and robustness of LLM sentence embeddings. SynTextBench is the pioneering effort in developing synthetic benchmarking methodologies for NLP, with a primary focus on sentence classification tasks and does not cover other NLP tasks (e.g. question answering, machine translation, summarization). By concentrating on the task, we have provided a solid foundation upon which future research can build. We believe that our work is a major step towards ensuring independent and sustainable auditing of LLMs.

References

Awasthi, P., Frank, N., and Mohri, M. On the existence of the adversarial bayes classifier. *Advances in Neural Information Processing Systems*, 34:2978–2990, 2021.

Baccianella, S., Esuli, A., and Sebastiani, F. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf.

Bhagoji, A. N., Cullina, D., and Mittal, P. Lower bounds on adversarial robustness from optimal transport. *Advances in Neural Information Processing Systems*, 32, 2019.

Biderman, S., Prashanth, U. S., Sutawika, L., Schoelkopf, H., Anthony, Q., Purohit, S., and Raf, E. Emergent and predictable memorization in large language models. *arXiv preprint arXiv:2304.11158*, 2023.

Blier, L. and Ollivier, Y. The description length of deep learning models. *Advances in Neural Information Processing Systems*, 31, 2018.

Blodgett, S. L. and Madaio, M. Risks of ai foundation models in education. *arXiv preprint arXiv:2110.10024*, 2021.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Bradley, M. M. and Lang, P. J. Affective norms for english words (anew): Instruction manual and affective ratings. Technical report.

Chiang, C.-H. and Lee, H.-y. On the transferability of pre-trained language models: A study from artificial datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 10518–10525, 2022.

Chuang, Y.-S., Dangovski, R., Luo, H., Zhang, Y., Chang, S., Soljačić, M., Li, S.-W., Yih, S., Kim, Y., and Glass, J. Diffcse: Difference-based contrastive learning for sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4207–4218, 2022.

Conneau, A. and Kiela, D. Senteval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 670–680, 2017.

Dadhich, A. and Thankachan, B. Opinion classification of product reviews using naïve bayes, logistic regression and sentiwordnet: Challenges and survey. In *IOP Conference Series: Materials Science and Engineering*, volume 1099, pp. 012071. IOP Publishing, 2021.

Dan, C., Wei, Y., and Ravikumar, P. Sharp statistical guaratees for adversarially robust gaussian classification. In *International Conference on Machine Learning*, pp. 2345–2355. PMLR, 2020.

Denecke, K. Using sentiwordnet for multilingual sentiment analysis. In *2008 IEEE 24th international conference on data engineering workshop*, pp. 507–512. IEEE, 2008.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. 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)*, pp. 4171–4186, 2019.

Esuli, A. Automatic generation of lexical resources for opinion mining: models, algorithms and applications. In *Acm sigir forum*, volume 42, pp. 105–106. ACM New York, NY, USA, 2008.

Esuli, A. and Sebastiani, F. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, 2006.

Esuli, A. and Sebastiani, F. Sentiwordnet: a high-coverage lexical resource for opinion mining. 2007.

Ethayarajh, K. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 55–65, 2019.

Ganguli, D., Hernandez, D., Lovitt, L., Askell, A., Bai, Y., Chen, A., Conerly, T., Dassarma, N., Drain, D., Elhage, N., et al. Predictability and surprise in large generative models. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1747–1764, 2022.

Gao, J., He, D., Tan, X., Qin, T., Wang, L., and Liu, T. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*, 2019.

Gao, T., Yao, X., and Chen, D. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910, 2021.

Gao, X., Zhang, Y., Galley, M., Brockett, C., and Dolan, W. B. Dialogue response ranking training with large-scale human feedback data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 386–395, 2020.

Giorgi, J., Nitski, O., Wang, B., and Bader, G. Declutr: Deep contrastive learning for unsupervised textual representations. 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)*, pp. 879–895, 2021.

Hauser, M. D., Chomsky, N., and Fitch, W. T. The faculty of language: what is it, who has it, and how did it evolve? *science*, 298(5598):1569–1579, 2002.

Hu, M. and Liu, B. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 168–177, 2004.

Husnain, M., Missen, M. M. S., Akhtar, N., Coustaty, M., Mumtaz, S., and Prasath, V. S. A systematic study on the role of sentiwordnet in opinion mining. *Frontiers of Computer Science*, 15(4):154614, 2021.

Johnson, R. A., Wichern, D. W., et al. *Applied multivariate statistical analysis*, volume 5. Prentice hall Upper Saddle River, NJ, 2002.

Ke, P., Ji, H., Liu, S., Zhu, X., and Huang, M. SentiLARE: Sentiment-aware language representation learning with linguistic knowledge. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6975–6988, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.567. URL https://aclanthology.org/2020.emnlp-main.567.

Khan, F. H., Qamar, U., and Bashir, S. Sentimi: Introducing point-wise mutual information with sentiwordnet to improve sentiment polarity detection. *Applied Soft Computing*, 39:140–153, 2016.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. Skip-thought vectors. *Advances in neural information processing systems*, 28, 2015.

Ko, C.-Y., Chen, P.-Y., Mohapatra, J., Das, P., and Daniel, L. Synbench: Task-agnostic benchmarking of pretrained representations using synthetic data. In *NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research*, 2022.

Lai, G., Xie, Q., Liu, H., Yang, Y., and Hovy, E. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, 2017.

Leskovec, J., Rajaraman, A., and Ullman, J. D. *Mining of massive data sets*. Cambridge university press, 2020.

Li, B., Zhou, H., He, J., Wang, M., Yang, Y., and Li, L. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9119–9130, 2020.

Li, H., Xu, M., and Song, Y. Sentence embedding leaks more information than you expect: Generative embedding inversion attack to recover the whole sentence. *arXiv preprint arXiv:2305.03010*, 2023.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

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

Mökander, J., Schuett, J., Kirk, H. R., and Floridi, L. Auditing large language models: a three-layered approach. *arXiv preprint arXiv:2302.08500*, 2023.

Ni, J., Abrego, G. H., Constant, N., Ma, J., Hall, K., Cer, D., and Yang, Y. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 1864–1874, 2022.

Nielsen, F. Å. A new anew: Evaluation of a word list for sentiment analysis in microblogs. In *Workshop on'Making Sense of Microposts: Big things come in small packages*, pp. 93–98, 2011.

Ohana, B. and Tierney, B. Sentiment classification of reviews using sentiwordnet. *Proceedings of IT&T*, 8, 2009.

Papadimitriou, I. and Jurafsky, D. Learning music helps you read: Using transfer to study linguistic structure in language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6829–6839, 2020.

Petridis, S. and Perantonis, S. J. On the relation between discriminant analysis and mutual information for supervised linear feature extraction. *Pattern Recognition*, 37 (5):857–874, 2004.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, 2016.

Rajpurkar, P., Jia, R., and Liang, P. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 784–789, 2018.

Rao, V. A., Anuranjana, K., and Mamidi, R. A sentiwordnet strategy for curriculum learning in sentiment analysis. In *Natural Language Processing and Information Systems: 25th International Conference on Applications of Natural Language to Information Systems, NLDB 2020, Saarbrücken, Germany, June 24–26, 2020, Proceedings 25*, pp. 170–178. Springer, 2020.

Rastogi, C., Ribeiro, M. T., King, N., and Amershi, S. Supporting human-ai collaboration in auditing llms with llms. *arXiv preprint arXiv:2304.09991*, 2023.

Ren, S., Deng, Y., He, K., and Che, W. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1085–1097, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1103. URL https://aclanthology.org/P19-1103.

Rezapour, R., Shah, S. H., and Diesner, J. Enhancing the measurement of social effects by capturing morality. In *Proceedings of the tenth workshop on computational approaches to subjectivity, sentiment and social media analysis*, pp. 35–45, 2019.

Shaikh, S., Cho, K., Strzalkowski, T., Feldman, L., Lien, J., Liu, T., and Broadwell, G. A. Anew+: Automatic expansion and validation of affective norms of words lexicons in multiple languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 1127–1132, 2016.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., Kluska, A., Lewkowycz, A., Agarwal, A., Power, A., Ray, A., et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, 2022.

Su, J., Cao, J., Liu, W., and Ou, Y. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*, 2021.

Thieme, A., Nori, A., Ghassemi, M., Bommasani, R., Andersen, T. O., and Luger, E. Foundation models in healthcare: Opportunities, risks & strategies forward. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–4, 2023.

Voita, E. and Titov, I. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 183–196, 2020.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2018.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

Warriner, A. B., Kuperman, V., and Brysbaert, M. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45:1191–1207, 2013.

Weidinger, L., Mellor, J., Rauh, M., Griffin, C., Uesato, J., Huang, P.-S., Cheng, M., Glaese, M., Balle, B., Kasirzadeh, A., et al. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*, 2021.

Whitney, W. F., Song, M. J., Brandfonbrener, D., Altosaar, J., and Cho, K. Evaluating representations by the complexity of learning low-loss predictors. *arXiv preprint arXiv:2009.07368*, 2020.

Wiggins, W. F. and Tejani, A. S. On the opportunities and risks of foundation models for natural language processing in radiology. *Radiology: Artificial Intelligence*, 4(4): e220119, 2022.

Zellers, R., Bisk, Y., Schwartz, R., and Choi, Y. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 93–104, 2018.

A. Appendix

A.1. Related Work and Background

Sentence representations. To obtain performant LLMs, learning universal sentence representations that capture rich information for various downstream NLP tasks without task-specific finetuning is an active research field and has also been studied extensively in the past years (Kiros et al., 2015; Conneau et al., 2017; Gao et al., 2019; Li et al., 2020; Su et al., 2021; Giorgi et al., 2021; Gao et al., 2021; Chuang et al., 2022). While learning to extract ideal sentence embeddings, (Gao et al., 2019; Li et al., 2020; Ethayarajh, 2019) have pinpointed the anisotropic behavior in the sentence embedding vector space as a reason behind sentence embeddings' poor capture of semantic information. To remedy the situation, Bert-flow (Li et al., 2020) and Bert-whitening (Su et al., 2021) transformed the sentence embedding distribution into an isotropic Gaussian distribution through normalizing flow and whitening post-processing. Through contrastive learning, SimCSE (Gao et al., 2021) and DiffCSE (Chuang et al., 2022) also achieved new state-of-the-art sentence embedding performance by promoting uniformity and alignment (Wang & Isola, 2020).

Evaluations of pretrained models. In evaluating the performance of LLMs, the current de facto evaluation paradigm is to utilize widely-used NLP benchmarks such as the General Language Understanding Evaluation (GLUE (Wang et al., 2018)/SuperGLUE (Wang et al., 2019)) benchmark, the Stanford Question Answering Dataset (SQuAD v1.1 (Rajpurkar et al., 2016)/v2.0 (Rajpurkar et al., 2018)), the Situations With Adversarial Generations (SWAG (Zellers et al., 2018)) dataset, the ReAding Comprehension from Examinations (RACE (Lai et al., 2017)) dataset, the Evaluation Toolkit for Universal Sentence Representations (SentEval (Conneau & Kiela, 2018)), BIG-Bench (Srivastava et al., 2022), etc. In many cases, these NLP benchmarks are supersets of datasets, e.g., GLUE is a collection of 9 datasets for evaluating natural language understanding systems, and SentEval is a collection of 7 Semantic Textual Similarity (STS) tasks and 7 transfer datasets that have partial overlap with GLUE. The heavy reliance on real-life tasks can be exemplified by broad literature. For example, Bert (Devlin et al., 2019) was evaluated on GLUE, SQuAD v1.1/2.0, SWAG; Roberta (Liu et al., 2019) was evaluated on GLUE, SQuAD v1.1/2.0, RACE; and T5 (Raffel et al., 2020) was evaluated on GLUE/SuperGLUE, SQuAD, CNN/Daily Mail abstractive summarization and WMT translation. HELM (Liang et al., 2022) proposes a holistic evaluation framework for language models that measures 7 metrics on 42 scenarios. However, when confronting the challenge of test-data leakage, to the best of our knowledge, there is no real-data-free evaluation method for NLP pretrained representations. In a recent literature (Ko et al., 2022), authors reported the validation loss (Val loss), minimum description length (MDL) (Blier & Ollivier, 2018; Voita & Titov, 2020), surplus description length (SDL) and ϵ-sample complexity (ϵSC) (Whitney et al., 2020) on class-conditional Gaussian distribution data as an effort to build task-agnostic evaluation baselines for pretrained representations in computer vision. Our proposed framework differs from this line of work in that we focus on the domain of natural language processing and we do not assume the data inputs are sampled from an idealized distribution. Instead, we create synthetic sentences and proxy tasks based on a lexical resource for LLM evaluation.

Sentiment lexicons. SentiWordNet 3.0 (Baccianella et al., 2010) is a lexical resource that provides sentiment information for each word in WordNet (Miller, 1995), a widely-used lexical database of English words and their relationships. SentiWordNet 3.0 is an improved version of SentiWordNet 1.0 (Esuli & Sebastiani, 2006), 1.1 (Esuli & Sebastiani, 2007), 2.0 (Esuli, 2008). SentiWordNet automatically assigns synsets of WordNet according to notions of "positivity", "negativity", and "neutrality". The sentiment scores of a synset are assigned on a scale from 0.0 to 1.0 and sum to 1, reflecting a fine-grained opinion-related word-level labeling. SentiWordNet has been used in a variety of natural language processing tasks, such as sentiment analysis (Denecke, 2008; Ohana & Tierney, 2009; Khan et al., 2016), opinion mining (Husnain et al., 2021; Dadhich & Thankachan, 2021), representation learning (Ke et al., 2020), and curriculum learning (Rao et al., 2020). Besides SentiWrodNet, other sentiment lexicons include Affective Norms for English Words (ANEW) (Bradley & Lang), Warriner lexicon (Warriner et al., 2013), a new ANEW (Nielsen, 2011), and ANEW+ (Shaikh et al., 2016). In this paper, we will demonstrate the use of sentiment lexicon with word-level labels in constructing synthetic datasets using SentiWordNet; however, the framework proposed herein can take any lexicon with word-level labels. We also envision our framework to benefit from a richer vocabulary and extend to other value lexicons like moral lexicons (Rezapour et al., 2019).

Robust Bayes optimal classifier. Despite the difficulty of characterizing the optimal classifier with the minimum loss for generic data, for data drawn from class-conditional Gaussian distribution, the explicit optimal strategy is given by Fisher's linear discriminant rule (Johnson et al., 2002; Petridis & Perantonis, 2004). Likewise, the optimal classification strategy can also be given for such data in the presence of input perturbations (Bhagoji et al., 2019; Dan et al., 2020). Let $\mathcal{N}(\mu, \Sigma)$ denote Gaussian distribution with mean μ and variance Σ. Generally, for binary classification problems with data pair (x, y) generated from a probability distribution $P_{\mu,\Sigma}$: $x|y = 1 \sim \mathcal{N}(\mu, \Sigma),\ x|y = -1 \sim \mathcal{N}(-\mu, \Sigma)$, the classifier that minimizes the adversarial loss (Awasthi et al., 2021) $\max_{x':\|x'-x\|\leq\epsilon} \mathbb{1}(f(x') \neq y)$, the robust Bayes optimal classifier (Bhagoji et al., 2019; Dan et al., 2020), is given by $\text{sign}(w_0^T x)$, where $w_0 = \Sigma^{-1}(\mu - z_\Sigma(\mu))$ and z_Σ is the solution of the problem

$$\arg\min_{\|z\|_2\leq\epsilon}(\mu - z)^T\Sigma^{-1}(\mu - z) \tag{1}$$

In the following sections, we will exploit robust Bayes optimal classifier in giving the explicit optimal classifier on whitened sentence embeddings and develop our theoretical groundings on top of it.

A.2. List of stop words

{‘must’, ‘meanwhile’, ‘among’, ‘same’, ‘you’, ‘formerly’, ‘already’, ‘take’, ‘he’, ‘thereupon’, ‘done’, ‘anyhow’, ‘almost’, ‘ca’, ‘regarding’, ‘will’, ‘mostly’, ‘say’, ‘again’, ‘forty’, ‘seemed’, ‘still’, ‘they’, “re’, ‘seem’, ‘latter’, ‘why’, ‘hers’, ‘thereby’, ‘themselves’, ‘your’, ‘nine’, ‘become’, ‘may’, ‘beyond’, ‘it’, ‘back’, ‘our’, ‘himself’, “m’, ‘via’, ‘we’, ‘seems’, ‘throughout’, ‘yourself’, ‘bottom’, ‘only’, ‘whereby’, ‘move’, ‘else’, ‘front’, ‘within’, ‘after’, ‘every’, ‘quite’, ‘hereby’, ‘now’, ‘since’, ‘became’, ‘herself’, ‘behind’, ‘any’, ‘those’, ‘used’, ‘indeed’, ‘’ve’, ‘first’, ‘moreover’, ‘ourselves’, ‘she’, ‘should’, ‘her’, ‘various’, ‘few’, ‘hundred’, ‘whoever’, ‘give’, ‘latterly’, ‘between’, ‘in’, ‘most’, ‘make’, ‘sixty’, ‘therefore’, ”’s”, ’hence’, ‘amount’, ‘otherwise’, ‘’m’, ‘’re’, ‘’s’, ‘are’, ‘could’, ‘along’, ‘ours’, ‘of’, ‘that’, ‘everywhere’, ‘during’, ‘his’, ‘then’, ‘fifty’, ‘namely’, ‘when’, ‘around’, ‘all’, ‘keep’, ‘these’, ‘’ll’, ‘third’, ‘being’, ‘thus’, ‘more’, “s’, ‘is’, ‘where’, ‘further’, ‘them’, ‘towards’, ‘next’, ‘and’, ‘a’, ‘does’, ‘here’, ‘ten’, ‘whom’, ‘except’, ‘myself’, ‘somehow’, ‘ever’, ‘enough’, ‘there’, ‘mine’, ‘other’, ‘so’, ‘hereupon’, ‘who’, ‘eight’, ‘one’, ‘hereafter’, ‘amongst’, ‘seeming’, ‘its’, ‘each’, ‘sometime’, ‘this’, ‘me’, “ll’, ‘until’, ‘him’, ‘because’, ‘many’, ‘anyway’, ‘part’, ‘from’, ‘have’, ‘over’, ‘to’, ”’re”, ’becomes’, ‘too’, ‘as’, ‘name’, ‘whence’, ‘whole’, ‘herein’, ‘everything’, ‘against’, ‘call’, ‘upon’, ‘both’, ‘i’, ‘whenever’, ‘across’, ‘anywhere’, ‘six’, ‘us’, ‘thereafter’, ‘also’, ‘former’, ‘whither’, ‘whose’, ‘such’, ‘really’, ‘was’, ‘’d’, ‘someone’, “ve’, ‘eleven’, ‘wherein’, ‘yours’, ‘by’, ‘their’, ‘beside’, ‘or’, ‘re’, ‘has’, ‘off’, ‘which’, ‘put’, ‘whether’, ‘per’, ‘four’, ‘whereafter’, ‘often’, ‘doing’, ‘had’, ‘out’, ‘some’, ‘fifteen’, ‘others’, ‘once’, ‘somewhere’, ‘either’, ‘besides’, ‘though’, ‘been’, ‘do’, ‘very’, ‘thru’, ‘go’, ‘please’, ‘sometimes’, ”’ll”, ’perhaps’, ‘whereupon’, ‘whatever’, ‘about’, ‘for’, ‘itself’, ‘thence’, ‘at’, ‘how’, ‘made’, ‘three’, ‘might’, ‘another’, ‘did’, ‘alone’, ‘elsewhere’, ‘toward’, ‘were’, ‘would’, ‘due’, ‘what’, ‘an’, ‘wherever’, ‘be’, ‘can’, ‘something’, ‘side’, ”’d”, ’with’, ”’m”, ’am’, ‘therein’, ‘into’, ‘through’, ”’ve”, ’everyone’, ‘on’, ‘my’, ‘even’, ‘own’, ‘see’, ‘several’, ‘two’, ‘afterwards’, ‘show’, “d’, ‘beforehand’, ‘nowhere’, ‘becoming’, ‘last’, ‘onto’, ‘the’, ‘yourselves’, ‘five’, ‘anyone’, ‘together’, ‘before’, ‘always’, ‘get’, ‘using’}

A.3. SentiWordNet 3.0 synsets

Each of the entries in SentiWordNet 3.0 has PosScore and NegScore denoting the positivity and negativity score assigned by SentiWordNet to the synset, and ObjScore is calculated by 1 - (PosScore + NegScore), denoting the neutrality score. When categorizing these words, we remove the sense number associated with the words and group words into individual word list based on the following criteria: for a word w,

- if PosScore $>$ NegScore, we categorize w into the positive word list;
- if PosScore $<$ NegScore, we categorize w into the negative word list;
- if PosScore $=$ NegScore $= 0$, we categorize w into the neutral word list.

We drop columns POS, ID, GLOSS in the examples for easier illustration. By performing the procedure on synsets in Table 4, we obtain a positive word list {able, living, accurate, concrete, active}, a negative word list {unfaithful, unable}, a neutral word list {acroscopic, straight}. In practice, we perform the procedure on SentiWordNet 3.0 and gather a positive word list with 23147 words, a negative word list with 26440 words, and a neutral word list with 154993 words.

Table 4: Examples of synsets in SentiWordNet 3.0.

SynsetTerms	PosScore	NegScore	SynsetTerms	PosScore	NegScore
able#1	0.125	0	unable#1	0	0.75
acroscopic#1	0	0	unquestioning#2	0.5	0.5
living#3	0.5	0.125	concrete#1	0.625	0.25
accurate#1	0.5	0	straight#5	0	0
unfaithful#4	0	0.5	active#5	0.5	0.125

A.4. More synthetic sentence examples

POSITIVE

- "convincingly gruesomely gruesomely convincingly deserve feeder exhaust exhaust debonaire stuffily stuffily anne_sexton wholeness wholeness rarefy conformable pretension pretension"
- "smarmily smarmily fairness covetously infuse soothing subtly subtly soothing"
- "precious grace the_right_way the_right_way absoluteness absoluteness"
- "personal_relation pleasurable sleekness cryptographically cryptographically correct delineate sink_in authenticated"
- "perfectibility lotus-eater shine shine health_care health_care pleasant-tasting"

NEGATIVE

- "counterrevolutionary apprehensive thunderclap unskilled unskilled thunderclap apprehensive cheat shanny shanny cheat counterrevolutionary smooth smooth decayed decayed imagine imagine loser unpicturesque unnaturalized unnaturalized unrelieved unrelieved unhewn"
- "unpleasant unpleasant mortal sympathetic dead dead choker nubbly fallout"
- 'jostling weka offend engorged fouled fouled engorged intermittence space impaction impaction space intermittence dishonesty disgustingly"
- "blindly blindly"
- "second_class criminal_possession lousiness nonextensile linanthus_dianthiflorus nonarbitrary regular foolishness stabbing"

A.5. Synthetic sentence generation details

During the construction of synthetic sentences, the probability p_e associated with the special token '<eos>' is determined by its frequency in the English Wikipedia corpus. For the remaining mass $1 - p_e$, p_n portion is assigned to new words, with its value picked following (Papadimitriou & Jurafsky, 2020), which is $p_n = 0.5$. Additionally, when there are no unpaired words in the stack (e.g., when drawing the starting token of the sentence, or when all the unpaired words are popped), we assign its probability $p_n(1 - p_e)$ to new words.

A.6. Histograms of synthetic datasets versus English Wikipedia corpus

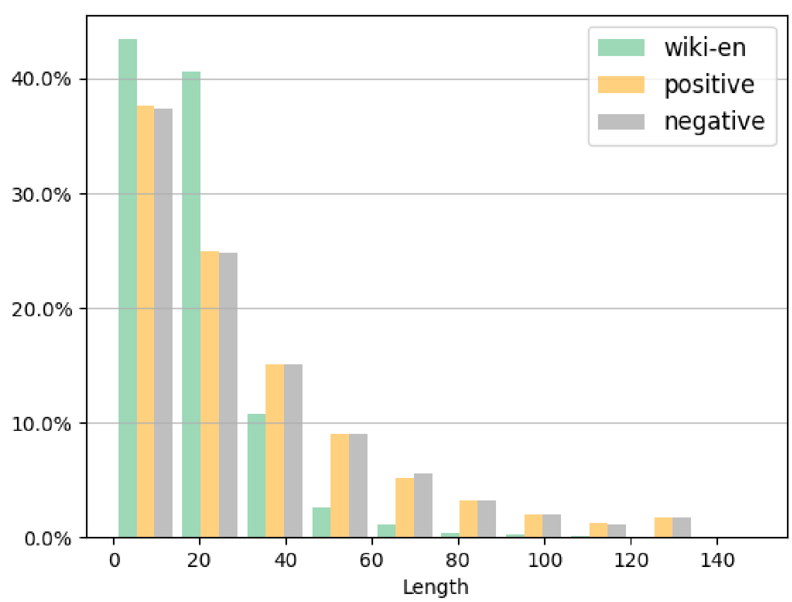


Figure 3: The histograms of sentence lengths in the English Wikipedia corpus (stop words removed) and the constructed synthetic corpus (positive/negative sentences).

A.7. Task feasibility

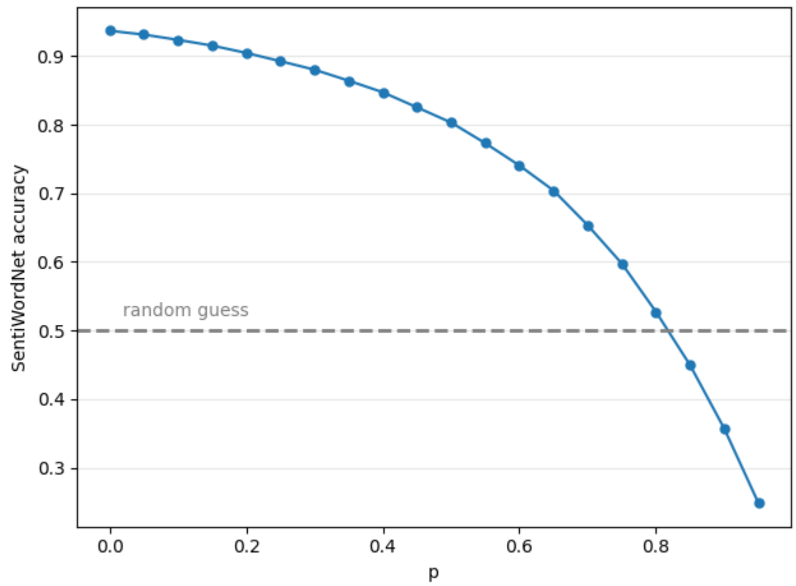
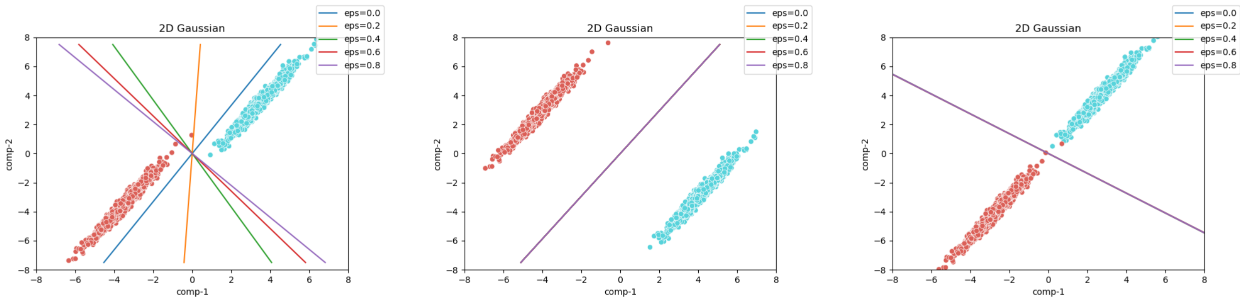


Figure 4: The reference accuracy given by SentiWordNet sentiment analysis. With an increasing mixing ratio p, the task becomes harder and the reference accuracy also shows a decreasing trend.

A.8. Robust Bayes optimal classifier and proofs

To motivate our findings, we first plot the Bayes optimal robust classifiers together with the Bayes optimal classifier in three 2D cases in Figure 5. From the plot, we see that as long as the direction of μ is in parallel to one of the two eigenvectors, the robust Bayes optimal classifiers would overlap with the Bayes optimal classifier.



(a) No alignment (b) $\mu \parallel v_1$ (c) $\mu \parallel v_2$

Figure 5: Three 2D examples of the Bayes optimal classifier and robust Bayes optimal classifiers with different magnitudes of expected perturbation ϵ. Figure 5(a) - no alignment between the mean vector μ and the eigenvectors. Figure 5(b) and Figure 5(c) - μ is parallel to the eigenvector corresponding to either of the two eigenvalues.

To generalize the result, we prove the following theorem that specifies a sufficient condition for all ϵ-robust Bayes optimal classifiers to overlap with each other (including $\epsilon = 0$, i.e. Bayes optimal classifier). Intuitively, if the ϵ-robust Bayes optimal classifiers overlap with the Bayes optimal classifiers, then there is no robustness-accuracy trade-off.

Result A.1. The ϵ-robust Bayes optimal classifiers overlap for all ϵ if the vector difference μ between the centers of the two gaussians lies completely within a degenerate subspace of the eigenspace of the covariance matrix, i.e. with eigenpairs $\{(\lambda_k, v_k), k \in [n]\}$, for $\forall\, i, j \in \{k : \lambda_k \neq 0, \mu^T v_k \neq 0\}$, $\lambda_i = \lambda_j = \lambda$.

Proof. Let $v_1, \ldots, v_n$ and $\lambda_1, \ldots, \lambda_n$ be the orthonormal eigenbasis and the corresponding eigenvalues of the covariance matrix Σ, then we have $\Sigma^{-1} = \sum_{i=1}^n \frac{1}{\lambda_i} v_i v_i^T$. Using (Dan et al., 2020), we see that the ϵ-robust classifier is given as $\operatorname{sign} {w^{\epsilon}}^\top x$, where $w^\epsilon = \Sigma^{-1}\left(\mu - z_\Sigma^\epsilon(\mu)\right)$ and

$$z_\Sigma^\epsilon(\mu) = \underset{\|z\| \leq \epsilon}{\arg\min} \|\mu - z\|_{\Sigma^{-1}}^2.$$

Let $\mu = \sum_{i=1}^n a_i v_i$ and we re-parameterize $z = \sum_{i=1}^n b_i v_i$. Then,

$$z_\Sigma^\epsilon(\mu) = \sum_{i=1}^n b_i^\epsilon v_i, \quad \text{where } b^\epsilon = \langle b_i^\epsilon \rangle_{i=1}^n = \underset{\sum_{i=1}^n b_i^2 \leq \epsilon^2}{\arg\min} \sum_{i=1}^n \frac{(a_i - b_i)^2}{\lambda_i}$$

By using the Lagrange multiplier γ_ϵ with first-order optimality condition, we see that $\forall\, i$

$$\frac{b_i^\epsilon - a_i}{\lambda_i} + \gamma_\epsilon b_i^\epsilon = 0 \iff \frac{a_i - b_i^\epsilon}{\lambda_i} = \gamma_\epsilon b_i^\epsilon \iff b_i^\epsilon = \frac{a_i}{1 + \lambda_i \gamma_\epsilon} \tag{2}$$

and $\sum_{i=1}^n \left(b_i^\epsilon\right)^2 \leq \epsilon^2$. In order for all the robust classifiers to overlap we need $w^\epsilon / \|w^\epsilon\|$ to the independent of ϵ. That is,

$$\frac{w^\epsilon}{\|w^\epsilon\|} = \frac{\sum_{i=1}^n v_i \frac{a_i - b_i^\epsilon}{\lambda_i}}{\sqrt{\sum_{i=1}^n \left(\frac{a_i - b_i^\epsilon}{\lambda_i}\right)^2}} = \frac{\sum_{i=1}^n \gamma^\epsilon b_i^\epsilon v_i}{\sqrt{\sum_{i=1}^n \left(\gamma^\epsilon\right)^2 \left(b_i^\epsilon\right)^2}} = \frac{\sum_{i=1}^n b_i^\epsilon v_i}{\sqrt{\sum_{i=1}^n \left(b_i^\epsilon\right)^2}} = \frac{\sum_{i \in S} b_i^\epsilon v_i}{\sqrt{\sum_{i \in S}^n \left(b_i^\epsilon\right)^2}},$$

where the S in the last equation denotes the set of indices for which $a_i \neq 0$. For $\forall\, i$ with $a_i = 0$, from equation 2, we clearly have $b_i^\epsilon = 0$.

The condition μ lies completely within a degenerate subspace of the eigenspace of Σ is equivalent to saying $\lambda_i = \lambda_j = \lambda$ for $\forall\, i, j \in S$. In this case, we see that for $\forall\, i \in S$,

$$\epsilon^2 \geq \sum_{i=1}^{n} (b_i^\epsilon)^2 = \sum_{i \in S} (b_i^\epsilon)^2 = \left(\frac{1}{1+\lambda\gamma_\epsilon}\right)^2 \sum_{i \in S} a_i^2,$$

so $\frac{1}{1+\lambda\gamma_\epsilon} \leq \epsilon \frac{1}{\sqrt{\sum_{i\in S} a_i^2}}$, $b_i^\epsilon \leq \frac{\epsilon}{\sqrt{\sum_{i\in S} a_i^2}} a_i$. So, we get $b_i^\epsilon = m_\epsilon \cdot a_i$ where $m_\epsilon = \min\left(1, \frac{\epsilon}{\sqrt{\sum_{i\in S} a_i^2}}\right)$

$$\frac{w^\epsilon}{\|w^\epsilon\|} = \frac{\sum_{i\in S} b_i^\epsilon v_i}{\sqrt{\sum_{i\in S}^{n} (b_i^\epsilon)^2}} = \frac{\sum_{i\in S} m_\epsilon a_i v_i}{m_\epsilon \sqrt{\sum_{i\in S} a_i^2}} = \sum_{i\in S} \frac{a_i}{\sqrt{\sum_{i\in S} (a_i)^2}} v_i,$$

which is independent of ϵ. $\square$

Result A.2. Consider the robust Bayes optimal classifier[1], f_ϵ, for P_{μ_1,μ_2,I_d} with class prior $\mathbb{P}(y=1)=\tau$, $\mathbb{P}(y=-1) = 1-\tau$, it is in the following form

$$f_\epsilon(x) = \text{sign}\left\{\left(x - \frac{\mu_1+\mu_2}{2}\right)^T \tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) - q/2\right\},$$

where $\tilde{\mu} = \frac{\mu_1-\mu_2}{2}$ and $q = ln\{(1-\tau)/\tau\}$. For any sample x, f_ϵ gives the lower bound on the decision margin δ

$$\begin{aligned}
&\left(x + \delta - \frac{\mu_1+\mu_2}{2}\right)^T \tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) - q/2 = 0 \\
\Leftrightarrow\quad & \delta^T \tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) = q/2 - \left(x - \frac{\mu_1+\mu_2}{2}\right)^T \tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) \\
\Rightarrow\quad & \|\delta\|_2 \geq \frac{|(x - \frac{\mu_1+\mu_2}{2})^T \tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) - q/2|}{\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2},
\end{aligned}$$

which then yields the worst-case bound

$$\|\Delta\|_2 = \min \|\delta\|_2 = \frac{|(x - \frac{\mu_1+\mu_2}{2})^T \tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) - q/2|}{\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2}.$$

Since the bound $\|\Delta\|_2$ is subject to the positions of two Gaussians, we scale the bound by the distance from Gaussian centers to the classifier. We note that, since the class are imbalanced, the distances from the two Gaussian centers to the classifier f_ϵ are different, i.e. $\frac{|\tilde{\mu}^T \tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) - q/2|}{\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2}$ and $\frac{|\tilde{\mu}^T \tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) + q/2|}{\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2}$, respectively. We hereby take their average as the scaling factor and obtain

$$\begin{aligned}
\|\bar{\Delta}\|_2 &= \frac{|(x - \frac{\mu_1+\mu_2}{2})^T \tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) - q/2|}{\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2} \frac{2\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2}{|\tilde{\mu}^T \tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) - q/2| + |\tilde{\mu}^T \tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) + q/2|} \\
&= \frac{2|(x - \frac{\mu_1+\mu_2}{2})^T \tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) - q/2|}{|\tilde{\mu}^T \tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) - q/2| + |\tilde{\mu}^T \tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) + q/2|}.
\end{aligned}$$

[1]Dobriban, E., Hassani, H., Hong, D. and Robey, A., 2020. Provable tradeoffs in adversarially robust classification. arXiv preprint arXiv:2006.05161.

A.9. Algorithm

Algorithm 1 Benchmarking LLMs using synthetic datasets (*SynTextBench*)

input Sentiment lexicons S, a range of difficulty levels P, an LLM g, threshold accuracy a_T.
output SynTextBench score that quantifies the robustness-accuracy performance.

Construct positive/negative/neutral word lists from sentiment lexicon S.
for p in P **do**
  Generate a synthetic binary classification task and obtain training set (x^{train}, y^{train}) and test set (x^{test}, y^{test}).
  Calculate transformation T_1 and T_{-1} from $z_1^{train} = \{g(x) \mid (x,y) \in (x^{train}, y^{train}), y = 1\}$ and $z_{-1}^{train} = \{g(x) \mid (x,y) \in (x^{train}, y^{train}), y = -1\}$.
  Transform training set and test set $\hat{z_1}^{train} = T_1(z_1^{train})$, $\hat{z_{-1}}^{train} = T_{-1}(z_{-1}^{train})$ and $\hat{z_1}^{test} = T_1(z_1^{test})$, $\hat{z_{-1}}^{test} = T_{-1}(z_{-1}^{test})$.
  Derive the Bayes optimal classifier f according to $\text{sign}(\tilde{\mu}^T(\hat{z} - \frac{\mu_1+\mu_2}{2}))$ based on $\hat{z_1}^{train}$ and $\hat{z_{-1}}^{train}$, i.e. $\mu_1 = \text{mean}(\hat{z_1}^{train})$, $\mu_2 = \text{mean}(\hat{z_{-1}}^{train})$.
  Read out the accuracy a of f on $\hat{z_1}^{test}$ and $\hat{z_{-1}}^{test}$, and calculate the average scale margin $\delta := avg(\|\bar{\Delta}_z\|_2)$ according to $\|\bar{\Delta}_z\|_2 = \frac{|(\hat{z} - \frac{\mu_1+\mu_2}{2})^T \tilde{\mu}|}{\|\tilde{\mu}\|_2^2}$ for correctly-classified sentence embeddings.
  Denote the accuracy and average margin pair on the task by (a_p, δ_p).
end for
Define a goodness function $s(a) = \frac{1}{|P|} \sum_{\{p \in P, a_p > a\}} \delta_p$, for $a \in \mathbb{R}[0,1]$.
SynTextBench score $= \int_{a_T}^{1} s(a)da$.

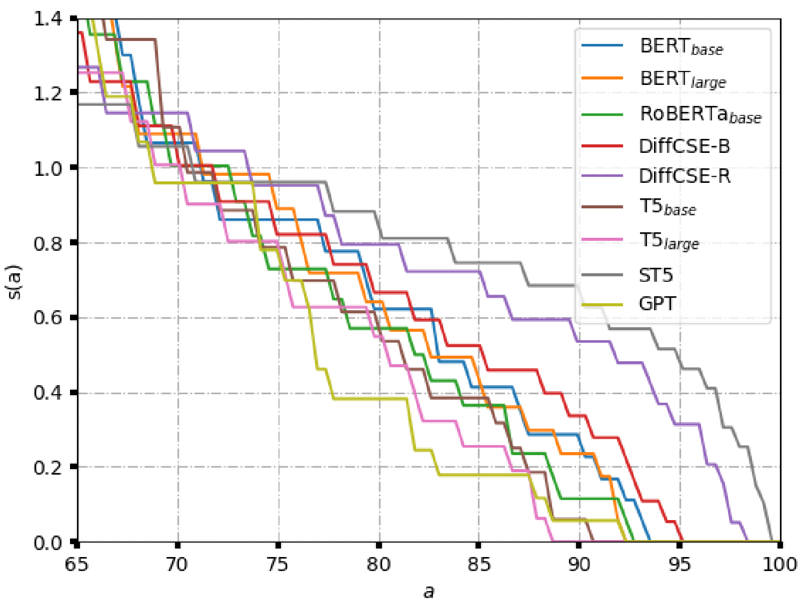


Figure 6: The goodness function $s(a)$ of nine pretrained language models. The SynTextBench score is calculated by the area under the curve.

A.10. Models

- $\text{BERT}_{\text{base}}$ and $\text{BERT}_{\text{large}}$ (Bidirectional Encoder Representations from Transformers (Devlin et al., 2019)) are encoder-only transformers pretrained with masked language model and next sentence prediction pre-training objectives with 110M and 340M parameters.
- $\text{RoBERTa}_{\text{base}}$ (Robustly Optimized BERT Pretraining Approach (Liu et al., 2019)) is a modification of BERT with 125M parameters that trained with dynamic masking, large mini-batches, a larger byte-level byte pair encoding, and removed the next sentence prediction objective.
- DiffCSE-B and DiffCSE-R (Difference-based Contrastive Learning for Sentence Embeddings (Chuang et al., 2022)) are $\text{BERT}_{\text{base}}$ and $\text{RoBERTa}_{\text{base}}$ models that further trained with difference-based contrastive learning.
- T5_{base} and T5_{large} (Text-to-Text Transfer Transformer (Raffel et al., 2020)) are encoder-decoder transformers with 223M and 738M parameters that casts all NLP tasks into a text-to-text problem.
- ST5 (Scalable sentence encoders from pre-trained text-to-text models (Ni et al., 2022)) is initialized by T5_{base} and trained by two-stage contrastive learning with 220M parameters.
- (GPT) DialogRPT (Dialog Ranking Pretrained Transformers (Gao et al., 2020)) is a decoder-only GPT-2 based transformer trained on vast human feedback data with 355M parameters. We use DialogRPT instead of GPT-2 since GPT-2 is not optimized for classification tasks while DialogRPT is fine-tuned with classification tasks.

A.11. Complete results

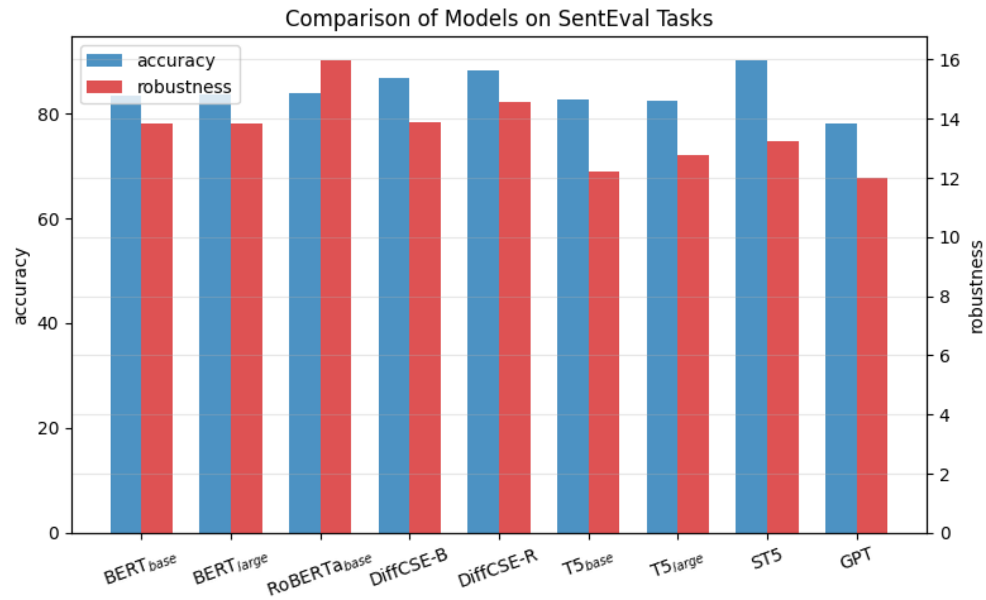


Figure 7: The accuracy and robustness (average percentage of perturbed words) performance of pretrained models on SentEval tasks.

Table 5: The detailed SentEval task performance. For STS tasks, we report Spearman's correlation (%), and for Transfer task, we report the standard accuracy (%).

	STS tasks							Transfer tasks								
Models	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	avg.
BERT$_{base}$	54.44	58.03	58.86	67.94	68.42	53.88	62.06	82.98	89.56	95.43	89.92	85.45	89.8	74.03	83.50
DiffCSE-B	68.88	76.21	73.88	79.76	78.84	75.51	67.70	82.2	88.11	95.44	91.03	84.46	88	75.71	86.81
BERT$_{large}$	53.33	56.86	56.23	63.43	66.69	54.43	58.06	85.96	89.59	96.43	90.96	89.13	91.8	73.16	83.68
T5$_{base}$	58.18	63.78	64.14	71.83	68.94	60.17	58.77	80.54	88.34	93.04	89.73	81.27	85.8	67.36	82.78
T5$_{large}$	58.34	62.59	63.50	71.36	67.88	59.67	58.02	79.31	86.86	93.53	90.43	80.72	82.8	68.75	82.36
RoBERTa$_{base}$	57.28	55.21	59.76	69.22	64.64	58.55	61.63	84.08	86.91	95.63	89.52	88.25	91.6	74.49	83.83
DiffCSE-R	69.77	78.70	76.08	81.75	80.86	81.17	70.34	84.75	90.99	95.2	89.75	87.92	89.4	77.28	88.19
GPT	44.16	23.99	34.73	40.78	55.11	41.05	43.65	81.08	88.53	92.81	87.87	86.6	93	70.49	78.01
ST5	74.32	82.83	81.50	86.14	85.95	86.04	79.76	85.88	91.81	94.4	91.09	90.88	95.8	74.26	90.17

Table 6: Pearson correlation comparison between real-data-free evaluation methods and the average accuracy on the real-life tasks included in Table 5. Since the smaller the Val loss, MDL, SDL and ϵSC, the better, we add a negative sign in front of them when calculating the Pearson correlation coefficient.

n	Name	BERT$_{base}$	DiffCSE-B	BERT$_{large}$	T5$_{base}$	T5$_{large}$	RoBERTa$_{base}$	DiffCSE-R	GPT	ST5	Pearson
	Reallife acc.	83.50	86.81	83.68	82.78	82.36	83.83	88.19	78.01	90.17	1.0
4096	Val loss	1.0e-06±1e-07	1.4e-06±3e-07	7.6e-07±5e-08	8.5e-08±1e-08	5.4e-08±9e-09	4.0e-06±3e-07	1.1e-06±8e-08	3.1e-03±8e-04	3.7e-03±5e-03	0.285±0.498
	MDL	5002±318	4755±129	5422±357	7318±119	6724±228	5396±181	4773±296	5604±366	4433±360	0.571±0.109
	SDL, ε=1	3090±318	2843±129	3510±357	5406±119	4812±228	3484±181	2861±296	3687±366	2514±368	0.570±0.110
	εSC, ε=1	3686±0	3686±0	3686±0	3686±0	3686±0	3686±0	3686±0	3686±0	3686±0	-
	SynTextBench	0.137±0.001	0.148±0.001	0.135±0.000	0.111±0.002	0.103±0.002	0.119±0.001	0.193±0.001	0.090±0.003	0.214±0.000	0.939±0.008
8192	Val loss	3.3e-06±3e-07	6.3e-04±9e-04	6.6e-04±9e-04	3.3e-07±9e-08	5.9e-04±8e-04	1.3e-05±1e-06	4.1e-06±2e-07	3.1e-02±1e-03	1.2e-03±5e-05	0.649±0.004
	MDL	8802±99	8687±260	10107±156	14664±464	14487±426	9801±489	8902±175	10001±291	7310±175	0.519±0.043
	SDL, ε=1	5262±99	5144±262	6564±155	11124±464	10944±426	6261±489	5362±175	6343±287	3766±175	0.509±0.043
	εSC, ε=1	7372±0	7372±0	7372±0	7372±0	7372±0	7372±0	7372±0	7372±0	7372±0	-
	SynTextBench	0.152±0.001	0.156±0.001	0.148±0.002	0.130±0.001	0.122±0.000	0.129±0.002	0.196±0.001	0.085±0.003	0.223±0.001	0.962±0.006
16384	Val loss	2.3e-03±2e-03	9.5e-04±7e-04	7.2e-04±1e-03	6.6e-04±9e-04	1.2e-03±9e-05	8.2e-04±1e-03	2.2e-03±2e-03	2.1e-01±3e-02	2.3e-02±9e-04	0.605±0.007
	MDL	15840±436	15253±455	18039±778	26004±879	25606±767	16629±117	15465±349	16794±440	11895±89	0.506±0.032
	SDL, ε=1	9266±429	8689±458	11477±786	19443±887	19040±767	10066±118	8891±365	8525±383	5153±93	0.425±0.021
	εSC, ε=1	14745±0	14745±0	14745±0	14745±0	14745±0	14745±0	14745±0	14745±0	14745±0	-
	SynTextBench	0.161±0.000	0.164±0.001	0.161±0.001	0.145±0.000	0.141±0.001	0.137±0.000	0.198±0.001	0.087±0.001	0.227±0.001	0.958±0.002
32768	Val loss	6.4e-03±8e-04	4.2e-03±2e-03	4.1e-03±3e-04	3.1e-02±1e-02	3.0e-03±7e-04	1.4e-02±2e-03	1.1e-02±1e-02	4.7e-01±2e-02	2.9e-01±1e-02	0.267±0.018
	MDL	27667±294	25793±898	29577±253	43955±1616	39692±1520	27151±33	27546±646	28930±471	21999±88	0.481±0.029
	SDL, ε=1	15417±282	13581±927	17367±252	31282±1860	27501±1518	14775±50	15214±489	9442±195	6076±106	0.311±0.008
	εSC, ε=1	29491±0	29491±0	29491±0	29491±0	29491±0	29491±0	29491±0	12139±0	12139±0	-0.044±0.000
	SynTextBench	0.170±0.001	0.169±0.000	0.173±0.001	0.158±0.001	0.156±0.000	0.140±0.001	0.202±0.000	0.092±0.001	0.230±0.000	0.934±0.002

A.12. Experimental details

When we calculate the correlation between real-data-free evaluation methods and real-life task robustness-accuracy performance, we need to aggregate two metrics - accuracy and robustness. For this purpose, we can obtain a ranking of the models according to the accuracy measure, R_1, and a ranking of the models according to the robustness measure, R_2. We aggregate two rankings by the simple and commonly-used mean aggregation[2] which yields the overall ranking of models based on accuracy-robustness performance, R_{ref}. On the other hand, we can obtain another ranking of models based on one of the real-data-free evaluation methods (e.g. Val loss, MDL, SDL, ϵSC, SynTextBench), R. Lastly, we calculate the Pearson correlation coefficient between R and R_{ref}.

Moreover, when we calculate the robustness measures, we only perform attacks on Transfer tasks as they are classification tasks where adversarial attacks are well-defined. Since we use the average percentage of perturbed words as the robustness indicator, we also excluded MPQA and TREC due to their short sentence lengths (MPQA and TREC average sentence lengths are 3.03 and 6.48, respectively). We list the robustness results in the following table:

Table 7: The robustness (average percentage of perturbed words) of pretrained representations on Transfer tasks.

Models	MR	CR	SUBJ	SST	MRPC	avg.
$\text{BERT}_{\text{base}}$	14.48	13.99	20.2	15.07	5.45	13.838
DiffCSE-B	14.46	14.7	18.64	15.19	6.39	13.876
$\text{BERT}_{\text{large}}$	14.3	14.22	19.87	15.46	5.26	13.822
T5_{base}	12.71	12.82	16.8	13.66	5.05	12.208
T5_{large}	13.67	14.28	16.93	13.82	5.17	12.774
$\text{RoBERTa}_{\text{base}}$	16.4	18.35	20.74	17.26	7.12	15.974
DiffCSE-R	15.72	16.07	18.53	16.82	5.68	14.564
GPT	12.53	13.11	15.75	13.52	5.17	12.016
ST5	13.6	13.08	18.36	14.22	6.9	13.232

We also list the ranking of models from different metrics in the following table.

Table 8: Ranking of models from different metrics at $n = 8192$.

Name	$\text{BERT}_{\text{base}}$	DiffCSE-B	$\text{BERT}_{\text{large}}$	T5_{base}	T5_{large}	$\text{RoBERTa}_{\text{base}}$	DiffCSE-R	GPT	ST5
Overall accuracy	6	3	5	7	8	4	2	9	1
STS accuracy	7	3	8	4	5	6	2	9	1
Transfer accuracy	5	6	2	8	9	4	3	7	1
Robustness	4	3	5	8	7	1	2	9	6
Val loss	8	4	3	9	5	6	7	1	2
MDL	7	8	3	1	2	5	6	4	9
SDL, ε=1	7	8	3	1	2	5	6	4	9
εSC, ε=1	5	5	5	5	5	5	5	5	5
SynTextBench	4	3	5	6	8	7	2	9	1

For example, to calculate SynTextBench correlation with robustness-and-accuracy performance, we calculate the Pearson correlation between (row "Overall accuracy" + row "Robustness") / 2 and "SynTextBench". To calculate SynTextBench correlation with robustness-and-STS accuracy performance, we calculate the Pearson correlation between (row "STS accuracy" + row "Robustness") / 2 and "SynTextBench". To calculate SynTextBench correlation with robustness-and-Transfer accuracy performance, we calculate the Pearson correlation between (row "Transfer accuracy" + row "Robustness") / 2 and "SynTextBench". We note that in all our results prior to Table 8, we always infer the correlation in individual runs before we take an average over all trials. Different from that, the rankings from Val loss, MDL, SDL, ϵSC, and SynTextBench in Table 8, are inferred from the average metric results over 3 trails for an easier illustration. Therefore, the ranking correlation suggested by the table might have some deviation from what is shown in Table 2.

[2]Wald, R., Khoshgoftaar, T.M. and Dittman, D., 2012, December. Mean aggregation versus robust rank aggregation for ensemble gene selection. In 2012 11th international conference on machine learning and applications (Vol. 1, pp. 63-69). IEEE.