On Robustness-Accuracy Characterization of Language Models using Synthetic Datasets

Ching-Yun Ko
MIT
cyko@mit.edu

Pin-Yu Chen & Payel Das
IBM
pin-yu.chen@ibm.com, daspa@us.ibm.com

Yung-Sung Chuang & Luca Daniel
MIT
yungsung@mit.edu, luca@mit.edu

Abstract

In recent years, language models (LMs) that were pretrained at scale on diverse data have proven to be a successful approach for solving different downstream tasks. However, 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 pre-trained LMs on sentence classification tasks. This approach allows for better characterization of the joint analysis on the robustness and accuracy of LMs without risking sensitive information leakage. It also provides a more controlled and private way to evaluate LMs that avoids overfitting specific test sets. Verified on various pretrained LMs, the proposed approach demonstrates promising high correlation with real downstream performance.

1 Introduction

In recent years, language models (LMs) have emerged, showcasing remarkable capabilities across a wide range of natural language processing (NLP) applications (Peters et al., 2018; Devlin et al., 2019; Yang et al., 2019; Raffel et al., 2020; Rae et al., 2021; Thoppilan et al., 2022; Hoffmann et al., 2022). While new opportunities present themselves with foundation models, they also bring forth potential risks and challenges (Bommasani et al., 2021; Blodgett & Madaio, 2021; Thieme et al., 2023; Biderman et al., 2023). For example, despite the unprecedented publicity of LMs and beliefs in their emergent abilities (Wei et al., 2022), some also argued the emergent abilities of LMs are a mirage (Schaeffer et al., 2023) and a change in metric choice can lead to a different conclusion. Recently, researchers have also expressed concerns about the potential for LMs to be trained on test sets (Li, 2023; Zhou et al., 2023; Golchin & Surdeanu, 2024; Oren et al., 2024; Xu et al., 2024). Even worse, private or held-out unpublished test sets may as well be vulnerable to data leakage through querying the LMs via APIs for evaluation purposes. Extraction attacks (Carlini et al., 2019; 2021), membership inference attacks (Hisamoto et al., 2020; Thomas et al., 2020; Mireshghallah et al., 2022), and generative embedding inversion attack (Li et al., 2023), caused by unintended memorization (Carlini et al., 2019; Shi et al., 2024) further deepened our concerns about test set contamination.

To address the caveat of test set contamination, in this paper, we aim to propose a new testbed for evaluating LMs with synthetic data. We link the design of the synthetic test set to two fundamental skills infants must master during language acquisition: identifying words and understanding linguistic structures (Frost et al., 2020). One intuitive approach is to generate labeled synthetic sentences using an existing generative LM and then evaluate

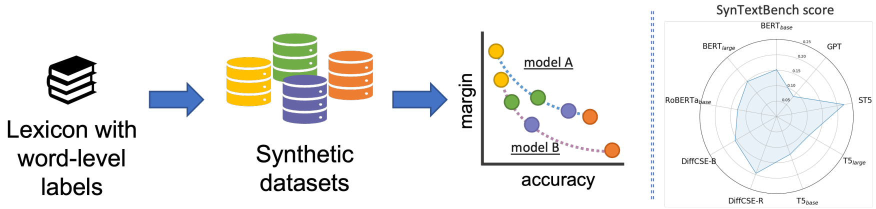


Figure 1: Overview of SynTextBench. SynTextBench generates a set of synthetic datasets from any given lexicon with word-level labels. We test the given LM on sentence-level tasks with these datasets and obtain robustness-accuracy characterization under a range of steerable task difficulties. For each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

LMs with the constructed test sets. By this, the generated sentences would harness language structural heuristics learned by the LM, and a decent probing result also requires the ability to distinguish words and their associated meanings, such as semantics. However, this workflow does not permit the active manipulation of synthetic task difficulties due to the limited level of interpretability (Zhang et al., 2023) and intrinsic bias of specific LMs (Acerbi & Stubbersfield, 2023). Motivated by the limitation, we explore an alternative route by entirely eliminating the reliance on LMs for test set generation. Specifically, we leverage 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 the lexicon, and construct sentences following 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 LMs on different levels of difficulty and complexity. Finally, we benchmark and quantify the ability of each LM on sentence classification tasks by comparing their performance on a set of our synthetic datasets with varying difficulty levels.

We dub our evaluation framework using synthetic data by *SynTextBench* and present the workflow in Figure 1, where we focus on benchmarking LM sentence embeddings in terms of their accuracy and robustness. By accuracy, we are interested in analyzing the linear separability of sentence representations rendered by different pretrained LMs. We note that in learning sentence embeddings, the go-to metrics are cosine distance or linear probing accuracy, both of which imply separability. By robustness, we refer to the decision margin on these sentence embeddings with respect to the optimal classification strategy. We derive both measures using only the constructed synthetic datasets, which allow for contamination-free benchmarking of LMs. SynTextBench complements existing benchmarks by providing a controlled environment to assess accuracy and robustness, which might be essential for reasoning and planning tasks. SynTextBench is designed as an extendable framework for the evaluation of language sentence representations that covers a range of controllable task difficulties.

Our **main contributions** are:

- We introduce *SynTextBench*, a novel theoretically-grounded framework to generate steerable synthetic datasets towards a holistic evaluation of LMs. The use of synthetic datasets alleviates the risk of test-data leakage and offers new tools for LM testing and auditing.
- SynTextBench provides a configurable lightweight testbed and a quantifiable metric for evaluating the robustness and accuracy of LMs 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 LMs 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 LMs.

2 Methodology

2.1 Why using synthetic datasets for LM evaluation?

To reduce the reliance on real-world data, we propose to build synthetic NLP tasks by generating synthetic sentences as model inputs at test time. This way, we no longer need to exchange sensitive private data or label-annotated data as test sets with LM APIs. In making a steerable and transparent evaluation framework for LMs, we first detail the desiderata of proxy tasks and the evaluation metric.

- Task substance: Tasks should test a pretrained LM'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 should be independent of the LM to be evaluated, in order to avoid biased evaluation, i.e., neither sentences nor labels should be given by an LM.
- 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 quantifiable metric that has a clear implication (e.g., 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 LM: (1) task difficulty would not be configurable (see more evidence in Appendix A.2), (2) the evaluation might be biased and favor the LM that generates the synthetic sentences or labels due to the intrinsic bias of each LM, and (3) any auditor without access to proprietary LMs or datasets cannot run independent evaluation.

In the following, we explain how we leverage sentiment lexicons, such as SentiWordNet 3.0, to create building blocks for our framework. Then, we put together building blocks and generate synthetic inputs to LMs by observing a nesting structure. We adjust the mixing ratio of ingredients in the recipe to simulate tasks of different difficulties. We depict this procedure in Figure 2. Finally, we will introduce our evaluation workflow and how we arrive at a quantifiable metric.

2.2 Constructing synthetic datasets and tasks

Word List. Building a synthetic task requires us to define the synthetic inputs to be used. Here, we utilize sentiment lexicons with word-level labeling. SentiWordNet labels the synsets of WORDNET (Miller, 1995) according to the notions of "positivity", "negativity", and "neutrality". Each of the entries in SentiWordNet has PosScore and NegScore denoting the positivity and negativity score, 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.

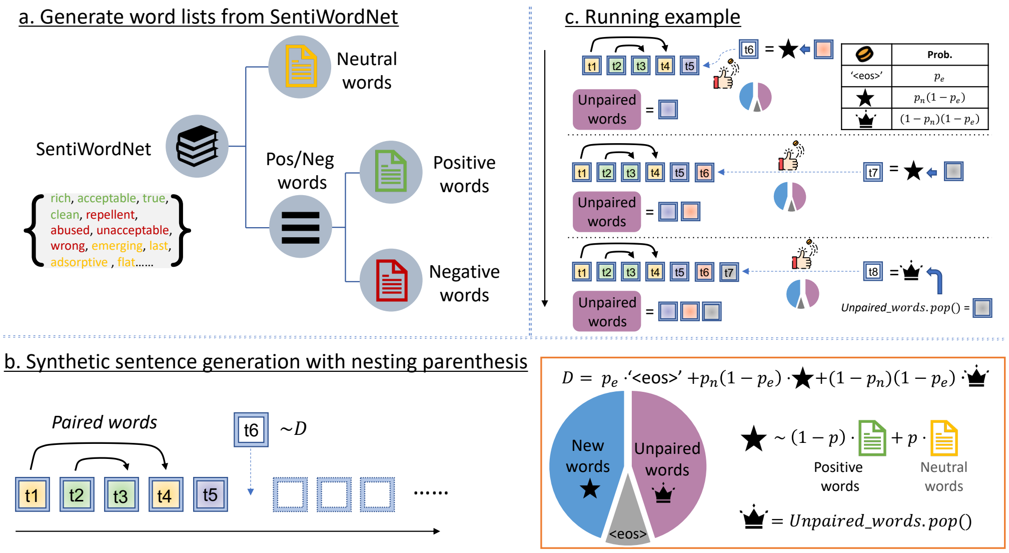

a. Generate word lists from SentiWordNet
SentiWordNet
rich, acceptable, true, clean, repellent, abused, unacceptable, wrong, emerging, last, adsorptive , flat......
Neutral words
Pos/Neg words
Positive words
Negative words
c. Running example
t1
t2
t3
t4
t5
t6
t7
t8
Unpaired words
=
Prob.
'<eos>'
p_e
$p_n(1-p_e)$
$(1-p_n)(1-p_e)$
Unpaired_words.pop() =
b. Synthetic sentence generation with nesting parenthesis
Paired words
~D
......
$D = p_e$ ·'<eos>' $+ p_n(1-p_e)$ · ★ $+(1-p_n)(1-p_e)$ ·
New words
Unpaired words
<eos>
★ ~ $(1-p)$ · Positive words + p · Neutral words
= Unpaired_words.pop()

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.

We give running examples in the Appendix A.3 for better understanding. 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. The same procedures can be applied to any sentiment lexicons with word-level labeling, which will result in different word lists. To this end, we created the word lists from SentiWordNet 3.0 as depicted in Figure 2(a).

Sentence structure. A recent literature (Papadimitriou & Jurafsky, 2020) explored the power of music and Java code in training models that transfer to NLP tasks. It further stated that, not only music and Jave code, non-linguistic artificial parentheses languages can also train LMs that yield substantial gains compared to random data when testing on natural language (Chiang & Lee, 2020; Ri & Tsuruoka, 2022; Papadimitriou & Jurafsky, 2023). Motivated by this, we follow one of the abstract structures, nesting parenthesis, when generating the synthetic sentences in our proxy tasks. The inclusion of the parenthesis is to guarantee we test for the linguistic structures, whose importance is repeatedly advocated in literature from both machine learning and cognitive science (Frost et al., 2020; Wilson et al., 2020; Manning et al., 2020). 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, the words are hierarchically nested, meaning the token to be paired with t_2, which is t_3 in our case, 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. With the created word list from above, we will now explain how to do sentence generation following the structure introduced. Let us revisit the case in Figure 2(b). Assume we want to generate a positive sentence (label $y = 1$), and we already generated the first five tokens $t_1 : t_5$ in the sentence with colors denoting the picked word. Now, to decide the next token, we sample t_6 from a mixing distribution D, where

$$D = p_e \cdot \text{`<eos>'} + p_n(1-p_e) \cdot \text{last\_unpaired\_word} + (1-p_n)(1-p_e) \cdot D_{\text{new}}. \quad (1)$$

To interpret distribution D, we realize that there are essentially 3 possible outcomes for the incoming t_6 token: (1) it can be the end of sentence indicator '<eos>', (2) it can be the popped token from the stack that stores the unpaired words, i.e., the last unpaired word, (3) it can be 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 probability density function (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} gives the PDF of neutral words, and f_{POS} gives the PDF of 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 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 essence, with the generated sentence, its label is determined by construction, which guarantees the **task independence** since the label is not given by an LM. 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. On the whole, by fixing a mixing ratio p, together with the fixed p_e and p_n given in the above, one synthetic dataset will be constructed as well as a resulting proxy sentiment classification task. By varying the mixing ratio p, a set of tasks with diverse difficulties can be created. In the Appendix A.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 LM or private data, and can be readily replicated by anyone with limited resources. Furthermore, we note that the construction of synthetic datasets and tasks described herein is also extendable to other lexicons and tasks by swapping the lexicon used for extracting word lists.

Lastly, we note that 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. We show the length profile of our synthetic data in Appendix A.5.

A workout example. Suppose we are to generate a synthetic task with difficulty $p = 0.7$. Now, to generate a positive sentence for the task, we consider sampling the first token t_1 following equation 1. At the first token, the mixing distribution $D = D_{\text{new}}$ since last_unpaired_word is an empty stack and the first token can not be the special token '<eos>'. We draw t_1 from $0.7 \cdot f_{\text{NEU}}(x) + 0.3 \cdot f_{\text{POS}}(x)$ and obtain t_1 = '*nice*'. At the second token, t_2 is sampled from the mixing distribution $D = p_e \cdot$ '<eos>' $+ p_n(1-p_e) \cdot$ last_unpaired_word $+ (1-p_n)(1-p_e) \cdot D_{\text{new}}$, where last_unpaired_word= ['*nice*'].

Discussions. The inclusion of parenthesis in our sentence structure guarantees we test for the linguistic structures but at the same time makes non-grammatical test sets. While grammar might be crucial in some NLP tasks that requires more advanced reasoning. For sentiment analysis, we believe it should not have a strong dependency on grammar (we exclude the scenario of negation which can be detected by a rule-based method). For example, the reviews "love love fantastic", "love fantastic love" and their word permutations should all be predicted as positive, regardless of their grammar. We support this intuition by additional experiment where we noticed that 86% of the labels given by Huggingface sentiment analysis pipeline on product reviews classification (Hu & Liu, 2004) remain the same after removing 284 stop words from the sentences and hence making them non-grammatical. We leave more details and sentence examples to Appendix A.6 and A.7.

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

```
Input: Sentiment lexicons $S$, a range of difficulty levels $P$, an LM $g$, 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 $(a_p, \delta_p)$.
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 $= \int_{a_T}^{1} s(a) da$.
```

2.3 Robustness-accuracy evaluation

Given an LM 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 linear classifiers (i.e., decision margins). 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. Nevertheless, 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) (also called variance-based methods by Falini (2022)). 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 an $\mathbb{R}^K$ vector space via $F_{:,1:k}^T S_{1:k,1:k}^{-1/2} z_y$. 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 sentences, 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 the optimization problem $\arg\min_{\|z\|_2 \leq \epsilon} (\mu - z)^T \Sigma^{-1} (\mu - z)$ (cf. Appendix A.1), we can prove that, when the covariance is an identity matrix, the class priors $\mathbb{P}(y = 1) = \tau$, $\mathbb{P}(y = -1) = 1 - \tau$, the perturbation radius ϵ, then the optimal classifier is given as simply f : $\text{sign}(w^T(\hat{z} - \frac{\mu_1+\mu_2}{2}) - q/2)$, 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 (i.e., $\tau = 1/2$), the robust Bayes optimal classifier overlaps with the Bayes optimal classifier. That is, the (robust) Bayes optimal classifier is plainly $\text{sign}(\tilde{\mu}^T(\hat{z} - \frac{\mu_1+\mu_2}{2}))$, which is independent of ϵ. We then use this given classifier to calculate the accuracy on the synthetic datasets. In fact, we prove in Appendix A.8 that, as long as $\tilde{\mu}$ 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 $\forall\ i, j \in \{k : \lambda_k \neq 0, \tilde{\mu}^T v_k \neq 0\}$, $\lambda_i = \lambda_j = \lambda$), the ϵ-robust Bayes optimal classifiers overlap for all ϵ. 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 $\|\tilde{\mu}\|_2$, 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 Appendix A.8. To this end, we have prepared sentence embeddings and specified the way of calculating decision margins induced by a robust Bayes optimal classifier. In the following, we will state the complete algorithm for characterizing robustness-accuracy performance of LMs using synthetic datasets.

2.4 SynTextBench score and algorithm

With Section 2.2 and Section 2.3, we now can simulate synthetic tasks of a configured level of difficulty and evaluate their accuracy and margin. In our benchmarking process, we essentially build on this foundation to generate a sequence 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 LM 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 (cf. Section 2.2). 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 LM'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, average margin) pair (a_p, δ_p) and draw one point on the margin-accuracy 2D plot as in Figure 1. We apply Algorithm 1 on various models and obtain a margin-accuracy curve for each model. Since we not only care about the curvature of the curve but also how the (accuracy, average 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 $\mathbb{R}[0, 1]$ in Line 10 to account for the span. By our definition, $s(a)$ will be a monotonically decreasing function (e.g., Appendix A.9) 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 in Line 11, 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 LM. In the later section, we will demonstrate the **metric informativeness** by measuring the correlation between SynTextBench scores and the average real-world sentence classification task performance.

3 Experiments

3.1 Setups

LMs. In the experiment, we will analyze the pretrained LMs predominantly considered by the sentence embedding literature Gao et al. (2021); Su et al. (2021); Chuang et al. (2022), and also larger models such as LLaMA and OPT (Touvron et al., 2023a;b; Zhang et al., 2022).. Specifically, we consider encoder models such as BERT$_{\text{base}}$, BERT$_{\text{large}}$ (Devlin et al., 2019), RoBERTa$_{\text{base}}$ (Liu et al., 2019), DiffCSE-B, DiffCSE-R (Chuang et al., 2022); encoder-decoder models such as T5$_{\text{base}}$, T5$_{\text{large}}$ (Raffel et al., 2020), ST5 (Ni et al., 2022); and decoder models such as DialogRPT (Gao et al., 2020)), LLaMA-7B, LLaMA-13B, LLaMA-30B (Touvron et al., 2023a), LLaMA-2-7B, LLaMA-2-13B (Touvron et al., 2023b), OPT-13B, OPT-30B (Zhang et al., 2022). 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 open-source implementation of the literature (Whitney et al., 2020) and fed the pretrained LMs 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.

Objectives. Through the experiments, our main aim is to verify the feasibility of making performance assessments of possible downstream tasks by real-data-free evaluation methods. To achieve this, we will compare the Pearson correlation coefficients of assessments given by different real-data-free evaluation methods with the performance on real-world tasks. Since SynTextBench is intended to inform the robustness-accuracy performance, we will report both the accuracy and robustness on real-world tasks for studying correlation. We use PWWS attack (Ren et al., 2019) through TextAttack, a Python framework for adversarial attacks in NLP, to generate attacks. Essentially, the attacker will perturb the inputs gradually by changing more and more words until the perturbation leads to a wrong classification result. Therefore, we report the average number of perturbed words in a successful attack as an indicator of the level of model robustness. We will also demonstrate how SynTextBench can be used to do attribute comparisons. Finally, as more attentions have been drawn to large LMs lately, we will also conduct an extended study on large LMs and include discussions on in-context learning performance on SynTextBench synthetic data. We defer experimental details to the appendix due to the page limit.

3.2 Performance evaluation and discussion

We evaluate encoder models listed in Section 3.1 by SynTextBench framework as well as on real-world sentence embedding tasks. Specifically, we simulated 20 synthetic datasets as described in Section 2.4 and obtained one goodness function $s(a)$ for each LM. We plot these functions together in Figure 7, from which the final SynTextBench score can be determined by definition. We refer readers to Appendix Table 7 for the exact numbers due to the page limit. To gauge the performance of these pretrained LMs on downstream real-world tasks, we evaluate the given models on SentEval (the Evaluation Toolkit for Universal Sentence Representations (Conneau & Kiela, 2018)) and show the detailed numbers in Appendix Table 6 and Figure 8. SentEval tasks include seven semantic textual similarity tasks (denoted by “STS tasks”), where results are given by the Spearman’s correlation with output range

Table 1: Correlation between 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±0.00
SynTextBench	**0.94±0.01**	**0.97±0.01**	**0.96±0.00**	**0.93±0.00**

Table 2: Aggregated correlation with real-data-free evaluation metric and the 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±0.03
εSC, ε=1	-	-	-
SynTextBench	**0.76±0.04**	**0.76±0.03**	**0.69±0.05**

$[-1, 1]$, 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.

Correlation with real-world tasks. To demonstrate the informativeness 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 the proposed SynTextBench. Since the smaller the baseline metrics are, the better, we add a negative sign in front of them when calculating the Pearson correlation coefficient. As we have the flexibility of generating synthetic datasets with various sizes (number of sentences), we compare four configurations $n = \{4096, 8192, 16384, 32768\}$. From Table 1, we observe that SynTextBench consistently gives scores highly correlated with real-world task accuracy, with correlation coefficients that are above 0.9. For the four baselines, the highest correlation ever achieved is when $n = 8192$ and evaluated by Val loss, 0.65. It is noteworthy that SynTextBench is also a stabler metric as substantiated by the smaller standard deviation.

Ablation on the nesting structure. To showcase the effect of the nesting structure, we see that no nesting structure is a special case of our proposed framework when $p_n = 0$ (cf. Equation 1). In Table 1, we have SynTextBench$(p_n = 0.5) = 0.97$. In comparison, we run the analysis for $p_n = 0$ and obtain SynTextBench$(p_n = 0) = 0.92$. In conclusion, SynTextBench, with both parameters, outperform the baselines by large margins. Between the two, SynTextBench with the imposed structure further improves the correlation.

Robustness implications. To understand how real-data-free evaluation methods correlate with real-world task robustness-accuracy performance, we further analyze the correlation with the robustness indicator, the average number of perturbed words, on Transfer tasks when $n = 8192$. We focus on these tasks as they 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 Appendix A.12 for more experimental 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. To be more precise, SynTextBench shows a coefficient of 0.76, whereas MDL and SDL are 0.64 and 0.60. Recall that accuracy results were aggregated from STS tasks and Transfer tasks. In Table 2, we also 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 still shows a stronger correlation compared to baselines, while having a slightly better correlation with Robustness-STS accuracy performance than Robustness-Transfer accuracy performance.

Case study on model comparisons. Besides having high correlation with real-world task performance, we show how SynTextBench can be used to make model comparisons. From Table 7, one sees that, the SynTextBench score of ST5 is significantly higher than that of T5 across all dataset sizes n, e.g., ST5's 0.223 vs. T5's 0.130 when $n = 8192$. This indicates contrastive fine-tuning is beneficial for improving sentence embeddings. This conclusion is in sync with the observations from real-world tasks, where we see ST5 yields both higher

Table 4: Pearson correlation comparison between the in-context learning accuracy on SynTextBench synthetic tasks and the average in-context learning accuracy on the real-world tasks of decoder models.

n	Name	LLaMA-7B	LLaMA-13B	LLaMA-30B	LLaMA-2-7B	LLaMA-2-13B	OPT-13B	OPT-30B	Pearson
	Reallife acc.	73.58	68.31	80.07	68.37	79.80	70.49	73.35	1.0
8192	SynTextBench	50.82	53.43	59.09	51.48	58.83	52.87	51.79	0.813

accuracy and robustness according to Table 7 and Table 11. Specifically, ST5 has an average accuracy of 90.17 and robustness 13.23, whereas T5 has an average accuracy of 82.78 and robustness 12.21.

3.3 Extended study on large LMs

Since SynTextBench focuses on the sentence embeddings of LMs, of which larger decoder models generally do not have better performance than smaller encoder models (Ethayarajh, 2019), we have given most of our analysis on encoder models in Gao et al. (2021). Here, to demonstrate the generality of SynTextBench to various LM types, we analyze more large decoder LMs such as LLaMA and OPT (Touvron et al., 2023a;b; Zhang et al., 2022).Similar to Table 1, we calculated the Pearson correlation coefficients between real-data-free evaluation methods and the accuracy of SentEval tasks in Table 3. According to the table, SynTextBench also gives scores highly correlated with real-world task accuracy on decoder models, with a correlation coefficient of 0.87. We refer readers to Appendix Table 9 for the complete results.

Table 3: Correlation between real-data-free evaluation metric and real-data accuracy.

Name	Pearson correlation
Val loss	0.80
MDL	-0.47
SDL, $\varepsilon = 1$	-0.55
εSC, $\varepsilon = 1$	-
SynTextBench	0.87

In-context learning Besides evaluating linear probing performance on our SynTextBench synthetic tasks, we also evaluate the few-shot in-context learning (ICL) performance on SentEval transfer tasks and SynTextBench synthetic task. We do not include STS tasks since they are typically measured by cosine distance, whose ICL prompts are less obvious to us. We also excluded TREC as we have not found proper prompts that could lead to reasonable accuracy. The instructions we give include two demonstrations with one demonstration for each class. For example, in CR (customer review), we use the instruction: "Answer the sentiment of the following review, either Positive or Negative. \n\nQ: We tried it out christmas night and it worked great .\nA: Positive\n\nQ: very bad quality .\nA: Negative\n\n".

In Table 4, we calculate the correlation between the ICL accuracy on SynTextBench synthetic tasks and the average ICL accuracy on subset SentEval tasks. We can see that the ICL accuracy on SynTextBench synthetic tasks shows strong correlation (above 0.8) with ICL accuracy on SentEval tasks. Future research will also be dedicated to investigate whether the success of SynTextBench can be explained by its ability to check the compositional features (e.g.induction head (Olsson et al., 2022)) of transformers.

4 Conclusion

In this paper, we have proposed SynTextBench, a novel framework for evaluating the accuracy and robustness of LM sentence embeddings. SynTextBench is a configurable real-date-free lightweight testbed that generates steerable synthetic language datasets and proxy tasks, avoiding the risk of test-data leakage. 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 such as question answering, machine translation, or summarization. By concentrating on this specific aspect, we have provided a solid foundation upon which future research can build.

Acknowledgments

This work was partially supported by the MIT-IBM Watson AI Lab and by the National Science Foundation.

References

Alberto Acerbi and Joseph M Stubbersfield. Large language models show human-like content biases in transmission chain experiments. In *Proceedings of the National Academy of Science*, volume 120, pp. e2313790120, 2023.

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

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 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.

Arjun Nitin Bhagoji, Daniel Cullina, and Prateek Mittal. Lower bounds on adversarial robustness from optimal transport. *Advances in Neural Information Processing Systems*, 32, 2019.

Kush Bhatia, Avanika Narayan, Christopher De Sa, and Christopher Ré. Tart: A plug-and-play transformer module for task-agnostic reasoning. *arXiv preprint arXiv:2306.07536*, 2023.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raf. Emergent and predictable memorization in large language models. *arXiv preprint arXiv:2304.11158*, 2023.

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

Su Lin Blodgett and Michael Madaio. Risks of ai foundation models in education. *arXiv preprint arXiv:2110.10024*, 2021.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

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

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, volume 267, 2019.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *USENIX Security Symposium*, volume 6, 2021.

Cheng-Han Chiang and Hung-yi Lee. Pre-training a language model without human language. *arXiv preprint arXiv:2012.11995*, 2020.

Cheng-Han Chiang and Hung-yi Lee. 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.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljačić, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. 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.

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

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 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.

A Dadhich and B Thankachan. 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.

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

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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.

Andrea Esuli. 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.

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

Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: a high-coverage lexical resource for opinion mining. 2007.

Kawin Ethayarajh. 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.

Antonella Falini. A review on the selection criteria for the truncated svd in data science applications. *Journal of Computational Mathematics and Data Science*, pp. 100064, 2022.

Rebecca LA Frost, Andrew Jessop, Samantha Durrant, Michelle S Peter, Amy Bidgood, Julian M Pine, Caroline F Rowland, and Padraic Monaghan. Non-adjacent dependency learning in infancy, and its link to language development. *Cognitive Psychology*, 120: 101291, 2020.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*, 2019.

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

Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and William B Dolan. 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.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 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.

Shahriar Golchin and Mihai Surdeanu. Time travel in LLMs: Tracing data contamination in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=2Rwq6c3tvr.

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

Sorami Hisamoto, Matt Post, and Kevin Duh. Membership inference attacks on sequence-to-sequence models: Is my data in your machine translation system? *Transactions of the Association for Computational Linguistics*, 8:49–63, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

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

Mujtaba Husnain, Malik Muhammad Saad Missen, Nadeem Akhtar, Mickaël Coustaty, Shahzad Mumtaz, and VB Surya Prasath. A systematic study on the role of sentiwordnet in opinion mining. *Frontiers of Computer Science*, 15(4):154614, 2021.

Richard Arnold Johnson, Dean W Wichern, et al. *Applied multivariate statistical analysis*, volume 5. Prentice hall Upper Saddle River, NJ, 2002.

Pei Ke, Haozhe Ji, Siyang Liu, Xiaoyan Zhu, and Minlie Huang. 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.

Farhan Hassan Khan, Usman Qamar, and Saba Bashir. Sentimi: Introducing point-wise mutual information with sentiwordnet to improve sentiment polarity detection. *Applied Soft Computing*, 39:140–153, 2016.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. *Advances in neural information processing systems*, 28, 2015.

Ching-Yun Ko, Pin-Yu Chen, Jeet Mohapatra, Payel Das, and Luca Daniel. Synbench: Task-agnostic benchmarking of pretrained representations using synthetic data. In *NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research*, 2022.

Kundan Krishna, Jeffrey P Bigham, and Zachary C Lipton. Does pretraining for summarization require knowledge transfer? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 3178–3189, 2021.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 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.

Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive data sets*. Cambridge university press, 2020.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 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.

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

Yucheng Li. Estimating contamination via perplexity: Quantifying memorisation in language model evaluation. *arXiv e-prints*, pp. arXiv–2309, 2023.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

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

Christopher D Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054, 2020.

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

Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. Quantifying privacy risks of masked language models using membership inference attacks. *arXiv preprint arXiv:2203.03929*, 2022.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 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.

Finn Årup Nielsen. 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.

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

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Yonatan Oren, Nicole Meister, Niladri S. Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. Proving test set contamination for black-box language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=KS8mIvetg2.

Isabel Papadimitriou and Dan Jurafsky. 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.

Isabel Papadimitriou and Dan Jurafsky. Pretrain on just structure: Understanding linguistic inductive biases using transfer learning. *arXiv preprint arXiv:2304.13060*, 2023.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202.

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

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 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.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 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.

Vijjini Anvesh Rao, Kaveri Anuranjana, and Radhika Mamidi. 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.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 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.

Rezvaneh Rezapour, Saumil H Shah, and Jana Diesner. 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.

Ryokan Ri and Yoshimasa Tsuruoka. Pretraining with artificial language: Studying transferable knowledge in language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7302–7315, 2022.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? *arXiv preprint arXiv:2304.15004*, 2023.

Samira Shaikh, Kit Cho, Tomek Strzalkowski, Laurie Feldman, John Lien, Ting Liu, and George Aaron Broadwell. 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.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=zWqr3MQuNs.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, 2022.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*, 2021.

Anja Thieme, Aditya Nori, Marzyeh Ghassemi, Rishi Bommasani, Tariq Osman Andersen, and Ewa Luger. 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.

Aleena Thomas, David Ifeoluwa Adelani, Ali Davody, Aditya Mogadala, and Dietrich Klakow. Investigating the impact of pre-trained word embeddings on memorization in neural networks. In *Text, Speech, and Dialogue: 23rd International Conference, TSD 2020, Brno, Czech Republic, September 8–11, 2020, Proceedings 23*, pp. 273–281. Springer, 2020.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Elena Voita and Ivan Titov. 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.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2018.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

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

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45:1191–1207, 2013.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

William F Whitney, Min Jae Song, David Brandfonbrener, Jaan Altosaar, and Kyunghyun Cho. Evaluating representations by the complexity of learning low-loss predictors. *arXiv preprint arXiv:2009.07368*, 2020.

Benjamin Wilson, Michelle Spierings, Andrea Ravignani, Jutta L Mueller, Toben H Mintz, Frank Wijnen, Anne Van der Kant, Kenny Smith, and Arnaud Rey. Non-adjacent dependency learning in humans and other animals. *Topics in cognitive science*, 12(3):843–858, 2020.

Ruijie Xu, Zengzhi Wang, Run-Ze Fan, and Pengfei Liu. Benchmarking benchmark leakage in large language models. *arXiv preprint arXiv:2404.18824*, 2024.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 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.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Comput. Surv.*, 56(3), oct 2023. ISSN 0360-0300. doi: 10.1145/3617680. URL https://doi.org/10.1145/3617680.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. Don't make your llm an evaluation benchmark cheater. *arXiv preprint arXiv:2311.01964*, 2023.

A Appendix

A.1 Related Work and Background

Sentence representations. To obtain performant LMs, 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 LMs, 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-world 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 LMs 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 LM 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 3.0; however, the framework proposed in this paper 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 $w_0 = \Sigma^{-1}\left(\mu - z_\Sigma(\mu)\right)$ and z_Σ is the solution of the convex problem

$$\arg\min_{\|z\|_2\leq\epsilon}(\mu - z)^T\Sigma^{-1}(\mu - z) \tag{2}$$

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 Generating synthetic datasets with a language model

To generate synthetic sentences with configurable difficulties with an LM, we reuse the word lists constructed in Section 2.2 and constrain the LM vocabulary to be within the word lists. Concretely, let V be the original tokenizer vocabulary, POS be the set of positive words, NEU be the set of neutral words, NEG be the set of negative words, and STOP be the set of stop words (see A.6), then we constrain the LM vocabulary to be $\bar{V} = \tilde{V} \cup \text{STOP}$, where $\tilde{V}$ composes of $p \times 100\%$ NEU $\cap\, V$ elements and $(1-p) \times 100\%$ POS $\cap\, V$ elements for positive sentence generations ($(1-p) \times 100\%$ NEG $\cap\, V$ elements for negative sentence generations). Similar to the use of the mixing ratio p in Section 2.2, we intend to create a set of tasks with diverse difficulties herein via varying p. We generate synthetic sentences by completing any of the starting tokens {”There”, ”I”, ”You”, ”She”, ”He”, ”It”, ”They”, ”The”}. We print some generated sentence examples below:

POSITIVE

- “She's a sweet and kind girl.”
- “The one thing that you have to do is look for other people.”
- “There are also a number of new content that have been rolled out in recent times.”
- “I had a lot of fun with this design.”
- “She was one of several of several hundred people in the group to speak out against the police and their use of force.”
- “You are very close to the truth.... if you are one of the first to see what is being done, that is very much a sign of an error.... you have to be very clear that it is a good thing that you are doing what you have to do......”

NEGATIVE

- “They were the worst of the worst.”
- “She has no other option.”
- “There's no question that the new and aggressive international community is headed for a bad start with its future in mind.”
- “They do not want to see you there.”
- “There's some real bad blood out there.”

- “I just want to make sure that we are talking about our state government.”

Discussions. Using LM-generated synthetic test sets, the rest of robustness-accuracy evaluation follows Section 2.3 and 2.4. We calculate the SynTextBench scores from LM-generated synthetic sentences and find that the Pearson correlation coefficient between these scores and the actual downstream task performance is 0.633±0.011. This is in contrast to the higher correlation coefficient of above 0.9 observed from the LM-free synthetic sentences discussed in Section 2.2, as shown in Table 1.

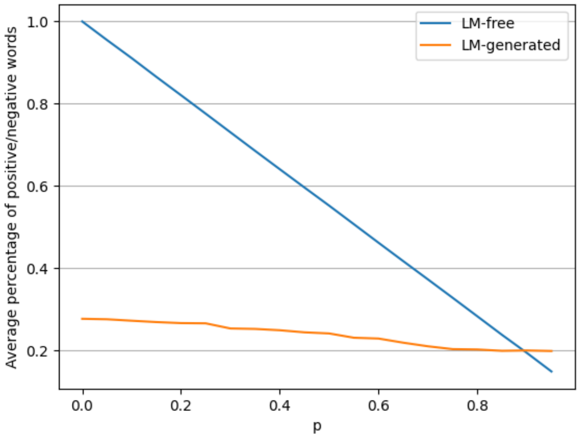


Figure 3: The average percentage of positive/negative words in the generated labeled positive/negative synthetic sentences. With an increasing mixing ratio p, we aim at configuring the task to be harder (data to be more mixed). While the percentage of positive/negative words does decrease in both LM-free synthetic sentences and LM-generated synthetic sentences, we have more control over LM-free generations in generating tasks at various difficulty levels (various y-axis values).

A.3 SentiWordNet 3.0 synsets

We drop columns POS, ID, GLOSS in the examples for easier illustration. By performing the procedure on synsets in Table 5, we obtain a positive word list {able, living, accurate, concrete, active}, a negative word list {unfaithful, unable}, a neutral word list {acroscopic, straight}.

Table 5: 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 Task feasibility

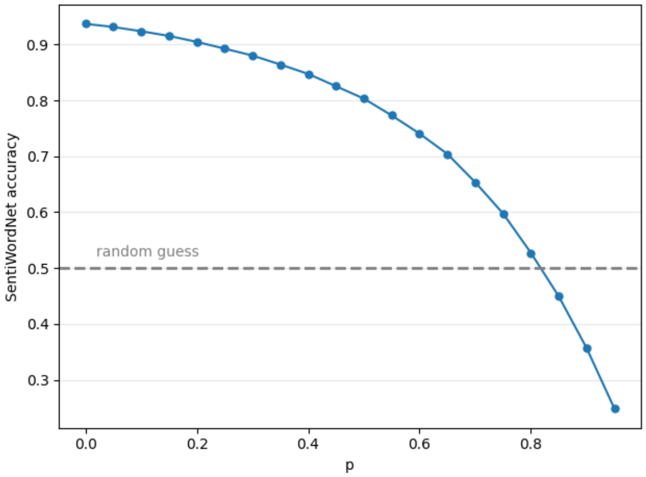


Figure 4: The reference accuracy given by SentiWordNet sentiment analysis. With an increasing mixing ratio p, the task becomes harder and the reference accuracy also shows a decreasing trend.

A.5 Histograms of synthetic datasets versus English Wikipedia corpus

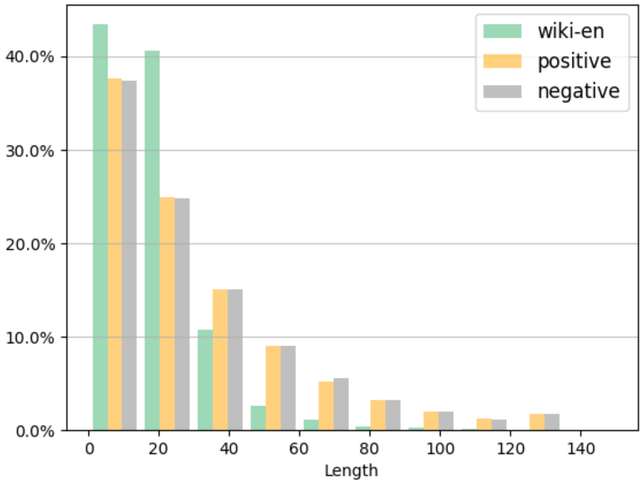


Figure 5: The histograms of sentence lengths in the English Wikipedia corpus (stop words removed) and the constructed synthetic corpus (positive/negative sentences).

A.6 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.7 Synthetic sentence examples and discussions

POSITIVE

- “perfectibility lotus-eater shine shine health_care health_care pleasant-tasting”
- “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

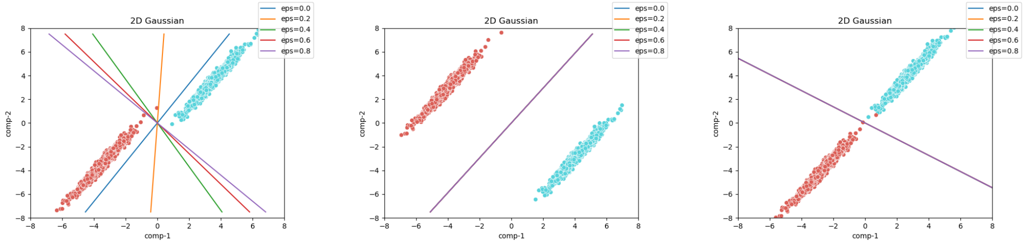
- “unpleasant unpleasant mortal sympathetic dead dead choker nubbly fallout”
- “counterrevolutionary apprehensive thunderclap unskilled unskilled thunderclap apprehensive cheat shanny shanny cheat counterrevolutionary smooth smooth decayed decayed imagine imagine loser unpicturesque unnaturalized unnaturalized 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”

Discussions. As we mentioned earlier in the paper, the inclusion of the parenthesis is to guarantee we test for the linguistic structures, whose importance is repeatedly advocated in literatures from both machine learning and cognitive science. Therefore, when building synthetic test for the linguistic structures, we also follow the parenthesis and thus have non-grammatical test sets.

We would like to motivate their use based on the following example of sentiment analysis in food reviews. Upon seeing the review “love love fantastic!” in a food review, a reasonable language model should recognize the entailed positive sentiment, even though the sentence is non-grammatical. In our framework, to test the other basic skill for language acquisition in a systematic and scalable manner, we put words associated with binary labels (positive and negative) in the synthetic sentence and test sentence embeddings of LMs in identifying the words for sentence classification. Related to our setups herein, Krishna et al. (2021) also studies a range of summarization tasks from nonsense documents, in which a task is also designed to classify whether there are keywords indicating positive or negative sentiments (Krishna et al. (2021), Figure 1). Additional evidence of the usage of non-grammatical sentences can be found in Bhatia et al. (2023), where authors also exploit non-grammartical synthetic sentence (Bhatia et al. (2023), Appendix A) for constructing Gaussian logistic regression problems in improving reasoning ability in LMs, which manifests the value of non-grammatical language in learning/testing basic skills. Our high correlation with real-world tasks further suggests that better understanding of the synthetic sentences indeed implies better performance on real tasks. By construction, our framework is not limited to sentiment analysis as one can readily change the base lexicon to test how LMs identify words describing other notions. For example, if we use the moral foundation lexicon, one can test how each LM identifies words that describe care, fairness, loyalty, authority, and sanctity.

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 6. 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 6: Three 2D examples of the Bayes optimal classifier and robust Bayes optimal classifiers with different magnitudes of expected perturbation ϵ. Figure 6(a) - no alignment between the mean vector μ and the eigenvectors. Figure 6(b) and Figure 6(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 $\Sigma^{-1} = \sum_{i=1}^{n} \frac{1}{\lambda_i} v_i v_i^T$. Following 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,

$$z^{\epsilon}_{\Sigma}(\mu) = \sum_{i=1}^{n} b_i^{\epsilon} v_i, \quad \text{where } b^{\epsilon} = \langle b_i^{\epsilon} \rangle_{i=1}^{n} = \operatorname*{arg\,min}_{\sum_{i=1}^{n} b_i^2 \le \epsilon^2} \sum_{i=1}^{n} \frac{(a_i - b_i)^2}{\lambda_i}$$

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

$$\frac{b_i^{\epsilon} - a_i}{\lambda_i} + \gamma_\epsilon b_i^{\epsilon} = 0 \iff \frac{a_i - b_i^{\epsilon}}{\lambda_i} = \gamma_\epsilon b_i^{\epsilon} \iff b_i^{\epsilon} = \frac{a_i}{1 + \lambda_i \gamma_\epsilon} \tag{3}$$

and $\sum_{i=1}^{n} \left(b_i^{\epsilon}\right)^2 \le \epsilon^2$. 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 3, 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 $\forall\ i \in S$,

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

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

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

which is independent of ϵ. □

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

$$\left(x + \delta - \frac{\mu_1 + \mu_2}{2}\right)^T \tilde{\mu}(1 - \epsilon / \|\tilde{\mu}\|_2) - q/2 = 0$$

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

$$\Rightarrow\ \|\delta\|_2 \ge \frac{|(x - \frac{\mu_1 + \mu_2}{2})^T \tilde{\mu}(1 - \epsilon / \|\tilde{\mu}\|_2) - q/2|}{\|\tilde{\mu}(1 - \epsilon / \|\tilde{\mu}\|_2)\|_2},$$

which then yields the worst-case bound

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

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

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

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

A.9 Goodness function

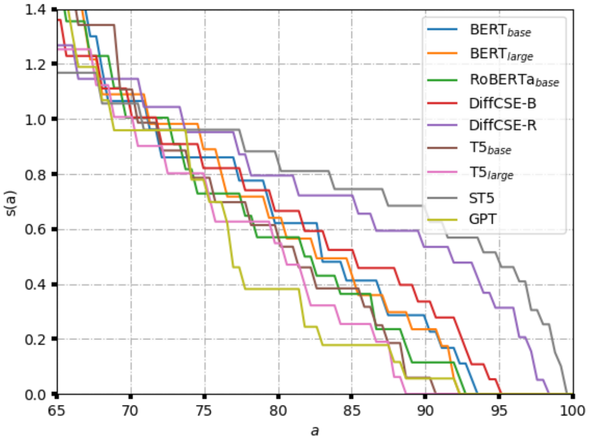


Figure 7: The goodness function $s(a)$ of nine pretrained LMs. The SynTextBench score is calculated by the area under the curve.

A.10 Complete results for Section 3.2

Table 6: The detailed SentEval linear probing 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.
$\text{BERT}_{\text{base}}$	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
$\text{BERT}_{\text{large}}$	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
$\text{RoBERTa}_{\text{base}}$	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 7: Pearson correlation comparison between real-data-free evaluation methods and the average linear probing accuracy on the real-world tasks included in Table 6. 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	$\text{BERT}_{\text{base}}$	DiffCSE-B	$\text{BERT}_{\text{large}}$	T5_{base}	T5_{large}	$\text{RoBERTa}_{\text{base}}$	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	0.968±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

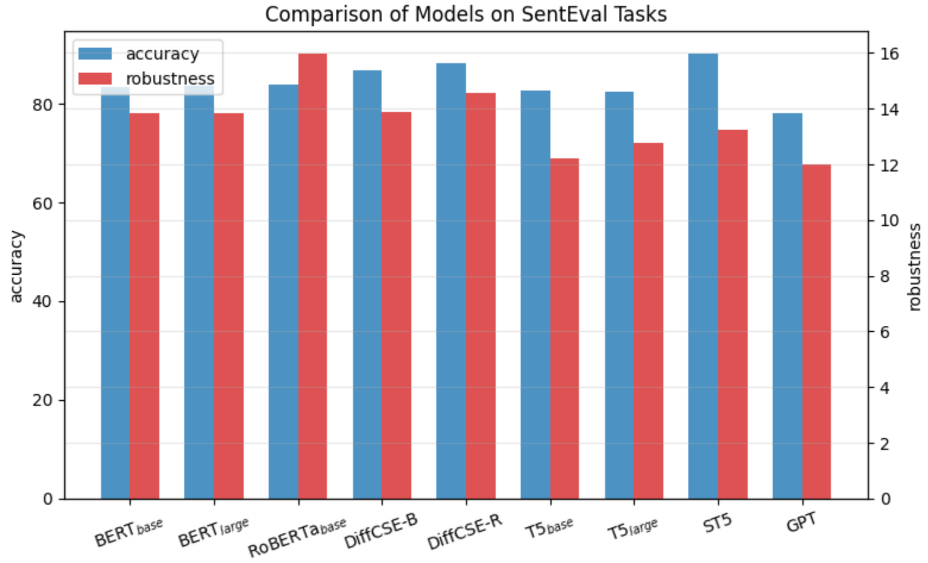


Figure 8: The accuracy and robustness (average number of perturbed words) performance of pretrained models on SentEval tasks.

A.11 Complete results for Section 3.3

Table 8: The detailed SentEval linear probing performance on decoder models. For STS tasks, we report Spearman's correlation (%), and for Transfer tasks, 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.
LLaMA-7B	10.51	9.68	5.85	2.60	5.87	15.58	15.01	71.20	75.87	87.59	81.94	77.59	62.80	64.12	64.55
LLaMA-13B	12.08	7.05	2.86	-0.84	7.38	3.50	10.93	70.88	77.06	88.04	81.53	76.77	64.00	63.19	63.78
LLaMA-30B	7.04	16.29	5.39	3.12	5.04	16.02	14.77	71.99	78.12	88.81	82.53	76.44	61.00	60.70	64.53
LLaMA-2-7B	11.95	22.85	10.85	16.31	44.42	20.13	47.17	91.07	91.95	97.30	89.22	94.78	96.80	67.88	76.13
LLaMA-2-13B	21.80	33.07	18.79	19.31	50.67	33.84	50.83	92.03	92.32	97.70	89.72	95.61	97.20	70.38	78.51
OPT-13B	24.20	40.78	24.91	25.75	56.70	39.44	51.32	91.23	92.45	97.13	89.28	95.00	96.80	72.58	79.72
OPT-30B	24.63	38.83	22.25	26.00	57.93	39.95	52.17	91.36	92.71	97.28	89.39	95.11	97.00	68.41	79.44

Table 9: Pearson correlation comparison between real-data-free evaluation methods and the average linear probing accuracy on the real-world tasks of decoder models. 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	LLaMA-7B	LLaMA-13B	LLaMA-30B	LLaMA-2-7B	LLaMA-2-13B	OPT-13B	OPT-30B	Pearson
	Reallife acc.	64.55	63.78	64.53	76.13	78.51	79.72	79.44	1.0
8192	Val loss	0.036141	0.149492	0.075583	0.000002	0.0	0.010351	0.00362	0.803
	MDL	8114.26	7434.78	6920.22	10331.5	9331.91	7874.07	7589.82	-0.466
	SDL, $\varepsilon = 1$	4435.77	3321.93	3090.58	6791.49	5791.91	4294.41	4035.95	-0.548
	εSC, $\varepsilon = 1$	7372	7372	7372	7372	7372	7372	7372	-
	SynTextBench	0.062	0.027	0.048	0.097	0.075	0.089	0.093	0.871

Table 10: The detailed subset SentEval in-context learning accuracy on decoder models.

	Transfer tasks						
Models	CR	MR	MPQA	SUBJ	SST2	MRPC	avg.
LLaMA-7B	85.35	90.49	74.34	48.97	88.47	53.86	73.58
LLaMA-13B	91.07	62.78	70.07	50.02	69.74	66.20	68.31
LLaMA-30B	91.97	92.60	83.77	50.01	95.83	66.26	80.07
LLaMA-2-7B	90.83	53.25	47.06	81.60	71.00	66.49	68.37
LLaMA-2-13B	91.84	91.92	80.26	52.73	95.55	66.49	79.80
OPT-13B	90.01	69.66	69.92	49.85	76.99	66.49	70.49
OPT-30B	90.78	82.04	63.56	50.00	87.10	66.61	73.35

A.12 Experimental details

When we calculate the correlation between real-data-free evaluation methods and real-world 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 number of perturbed words by PWWS attacks (Ren et al., 2019) 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). PWWS attacks focus on the text adversarial example generation that could guarantee little semantic shifting and therefore rarely cause ground truth label change (also lexical and grammatical correctness). To meet the semantic constraint, PWWS replaces words in the input texts with synonyms and replace named entities (NEs) with similar NEs to generate adversarial samples. Synonyms for each word can be found in WordNet, a large lexical database for the English language. NE refers to an entity that has a specific meaning in the sample text, such as a person's name, a location, an organization, or a proper noun. Replacement of an NE with a similar NE imposes a slight change in semantics but invokes no lexical or grammatical changes.

We list the robustness results in the following table:

Table 11: The robustness (average number of perturbed words) of pretrained representations on Transfer tasks.

Models	MR	CR	SUBJ	SST	MRPC	avg.
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
BERT$_{\text{large}}$	14.3	14.22	19.87	15.46	5.26	13.822
T5$_{\text{base}}$	12.71	12.82	16.8	13.66	5.05	12.208
T5$_{\text{large}}$	13.67	14.28	16.93	13.82	5.17	12.774
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 12: Ranking of models from different metrics at $n = 8192$.

Name	BERT$_{\text{base}}$	DiffCSE-B	BERT$_{\text{large}}$	T5$_{\text{base}}$	T5$_{\text{large}}$	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 "Ro-

[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.

bustness") / 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 12, 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 12, 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.