PTEB: Towards Robust Text Embedding Evaluation via Stochastic Paraphrasing at Evaluation Time with LLMs

Manuel Frank
Department of Computer Science
Munster Technological University
Manuel.Frank@zohomail.eu

Haithem Afli
Department of Computer Science
Munster Technological University
Haithem.Afli@mtu.ie

Abstract

Current sentence embedding evaluations typically rely on static test beds like the Massive Text Embedding Benchmark (MTEB). While invaluable, repeated tuning on a fixed suite can inflate reported scores and obscure real-world robustness. We introduce the Paraphrasing Text Embedding Benchmark (PTEB), a dynamic protocol that stochastically generates meaning-preserving paraphrases at evaluation time and aggregates results across multiple runs. Using a cost-efficient LLM-based method grounded in gold ratings and human validation, we show that LLMs generate token-diverse but semantically preserving paraphrases. Across 7 MTEB tasks, we validate our hypothesis that the performance of sentence encoders is sensitive to changes in token space even when semantics remain fixed. We also observe that smaller models are not disproportionately affected relative to larger ones. Our results are statistically robust over multiple runs spanning 20 datasets and 25 languages. More generally, we aim to propose a new evaluation paradigm in NLP that relies less on static, pre-defined benchmarks but shifts towards dynamic, stochastic evaluation leveraging eval-time compute. We make the code to run PTEB publicly available.[1]

1 Introduction

Text embeddings have become a cornerstone of Natural Language Processing (NLP), enabling a wide range of downstream tasks. Traditionally, embeddings are evaluated on static benchmarks like the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2023), a de facto standard. More recently, MTEB has been extended to the Massive Multilingual Text Embedding Benchmark (MMTEB) (Enevoldsen et al., 2025), covering more than 250 languages, and RTEB[2], a retrieval benchmark with private test sets. However, benchmarks like MTEB need to be maintained and further developed (Chung et al., 2025).

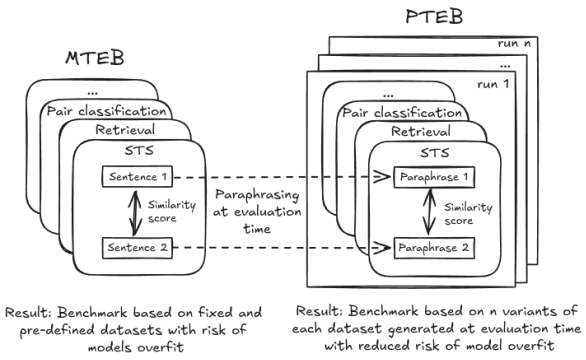

Figure 1: PTEB paraphrases data at evaluation time to measure embedding model performance.

A persistent challenge is that static benchmarks saturate over time. Although this saturation partly reflects genuine progress, it can also be attributed to overfitting. This arises from at least two factors (Liang et al., 2025): 1. *Data contamination* which refers to the inclusion of benchmark items within training data, enabling models to exploit prior exposure rather than demonstrate authentic modelling capabilities (also known as data leakage). 2. *Biased overtraining* which involves deliberate allocation of training resources to the domains of anticipated benchmarks, resulting in models with uneven performance that excel in benchmarked domains while underperforming elsewhere. Overall, more robust embedding benchmarks are needed (Goel et al., 2025). So far, the only solutions have been temporary through private test sets or the release of new or updated benchmarks.

To address this issue, we argue that model capabilities should be evaluated on dynamically generated variations of evaluation tasks at evaluation time rather than predefined static benchmarks. We therefore introduce the Paraphrasing Text Embedding Benchmark (PTEB), which generates para-

[1] https://github.com/ThisIsManuel/pteb
[2] https://huggingface.co/blog/rteb

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 2832–2851

phrases at eval time, creating a set of semantically equivalent but textually distinct problem instances that better approximates diverse real-world applications. While MTEB targets maximal language and task coverage, PTEB is a complementary, statistically robust protocol that stress-tests semantic invariance via multi-run eval-time paraphrasing.

Generative LLMs have been shown to be effective paraphrasers (Wahle et al., 2022), e.g., for data augmentation (Dai et al., 2023). Frank and Afli (2025) and Thirukovalluru and Dhingra (2025) used LLMs to generate variations of the input text, e.g., by paraphrasing, for data augmentation at runtime to improve the performance of embeddings.

Prior research on robustness and benchmarking has mainly relied on algorithmic transformations, masked language modelling (MLM), or classical word embedding techniques. For example, `nlpaug` (Ma, 2019) provides a general-purpose augmentation toolkit with methods such as backtranslation, synonym replacement, and embedding-based substitutions (e.g., using Word2Vec (Mikolov et al., 2013)), while TextAttack (Morris et al., 2020) systematises adversarial attacks through synonym replacments, character edits, and MLM substitutions, supporting both benchmarking and adversarial training. Building on these ideas, TextFlint (Wang et al., 2021b) introduced a multilingual robustness framework combining universal and task-specific transformations including adversarial attacks, all validated through human evaluation.

More recently, Yang et al. (2023) published an article on benchmark contamination and evasion of contamination detection methods. They demonstrated the fragility of existing evaluations for generative models, showing that rephrased benchmark samples can bypass standard decontamination methods and artificially inflate the performance of a 13B parameter model to GPT-4 levels, thus undermining benchmark trustworthiness. Whereas Yang et al. (2023) use paraphrasing to diagnose weaknesses of static benchmarks for generative LLMs, we employ eval-time paraphrasing to construct contamination-resistant evaluations for embedding models. Li et al. (2025) address similar concerns using semantic graph manipulation to generate hard negatives by altering semantics, though limited to English. Existing work has therefore emphasised robustness testing, augmentation, contamination detection or semantic manipulation, but remains fundamentally constrained by static transformation rules, MLM, classical embeddings, and a focus on generative models or English corpora.

Concurrent work by Goel et al. (2025) proposes SAGE to analyse robustness (among other dimensions) through rules-based transformations such as synonym replacement evaluated on 3 English datasets. Our approach complements this by testing robustness to richer LLM-generated paraphrases on 20 datasets with 25 languages across six random seeds validated by LLM-judges and humans.

Our contribution is to employ generative LLMs as paraphrasers at eval-time, yielding a stochastic, dynamically moving benchmark that continuously produces novel test variants. This extends robustness testing from predefined perturbations to rich, stochastic evaluation runs, specifically targeting embedding models across diverse NLP tasks. More broadly, our goal is to shift the focus of the NLP community from static datasets and offline augmentation to stochastic evaluation-time benchmarks powered by modern generative models.

The PTEB architecture is visualised in Figure 1. We make the code to run PTEB publicly available.[3]

Dataset	Word count
BIOSSES (Soğancıoğlu et al., 2017)	22.8
SICK-R (Marelli et al., 2014)	9.6
STS12 (Agirre et al., 2012)	10.9
STS13 (Agirre et al., 2013)	8.9
STS14 (Agirre et al., 2014)	9.2
STS15 (Agirre et al., 2015)	10.6
STS17 (Cer et al., 2017)	8.7
STS22 (Chen et al., 2022)	461.0
STSB (Cer et al., 2017)	9.9
Average	61.1

Table 1: Average word count per input text for English STS datasets.

2 Method

Our method is illustrated in Figure 2. We apply generative LLMs in two distinct roles: First, as an LLM judge to select a paraphrase model. Second, as a paraphraser generating paraphrases during evaluation. To evaluate and select the LLM judge, we use sentence pairs from the MTEB STS datasets. Without requiring human evaluations or direct validation of the generated paraphrases, our method anchors PTEB in scientific rigor by grounding it in STS gold scores. Table 1 lists the STS datasets that we use and shows that the STS

[3] https://github.com/ThisIsManuel/pteb

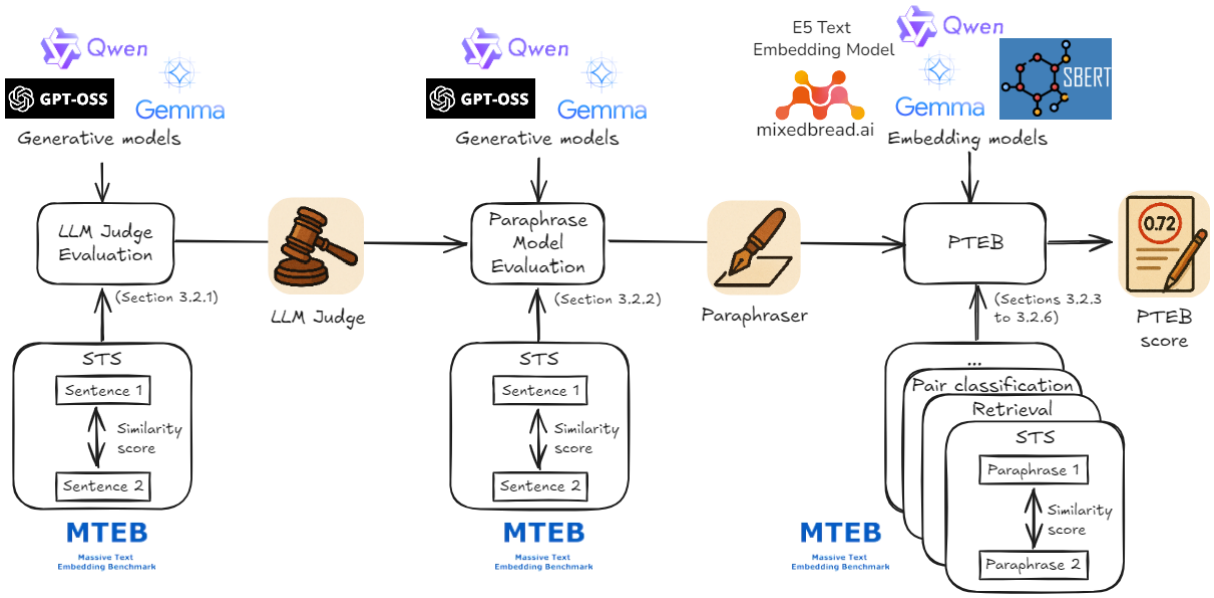

Figure 2: MTEB STS data is used to evaluate different generative LLMs for their STS rating capability. Using the selected LLM judge, generative LLMs are evaluated for paraphrasing MTEB STS data. Finally, the best paraphrase model is used to paraphrase MTEB datasets at eval-time generating a dynamic benchmark for embedding models.

datasets cover a range of different text lengths ranging from short sentences to STS22's paragraphs with an average length of 461 words. Throughout this paper, cosine similarity is applied as the similarity metric between embeddings. Generally, we use the same metrics as MTEB, for example, Spearman's rank correlation for STS (see Table 13 in Appendix A for a list of all metrics per task/dataset). All averages are reported as macro-averages; for robust effect size estimation we additionally calculate the Hodges-Lehmann (HL) estimator (Hodges and Lehmann, 1963) with 95% confidence intervals (CI) for selected analyses.[4]

1. LLM Judge Evaluation. The MTEB STS datasets provide similarity ratings on bounded scales, obtained through annotation methods like Amazon Mechanical Turk. We evaluate generative LLMs based on their ability to assess the semantic similarity of sentence pairs vs. these STS gold ratings in order to select an LLM judge model.

2. Paraphrase Model Evaluation. Next, we apply generative LLMs to paraphrase the MTEB STS datasets. The LLM judge selected in the previous step rates the semantic similarity of the paraphrases on a scale from 0 to 5. Based on semantic similarity, (normalised) edit distance (ED)[5], and runtime, we select the paraphrase model ("paraphraser"). We include the ED as a criterion to avoid textually overly similar paraphrases compared to the original text (larger ED is better).

3. Embedding Model Evaluation on PTEB. Finally, we apply the selected paraphraser to rephrase MTEB datasets at eval-time generating a dynamic benchmark. This allows us to evaluate various embedding models on the paraphrases and compare their scores with the original MTEB datasets. For details on all datasets incl. the metrics used see Table 13 in Appendix A. We run PTEB using $n = 6$ different paraphrasing seeds since single-run estimates can be unreliable (Reimers and Gurevych, 2017, 2018; Dror et al., 2019).

Given that steps 1 and 2 of our method rely on English datasets, we validate a subset of the paraphrases generated by the selected paraphrase model for non-English datasets using human raters. For selected languages, we randomly select 50 sentence pairs each and collect 2 human ratings per pair. We established an a priori acceptability threshold of ≥ 3.00 for the mean human similarity ratings (scale: 0 to 5). See Appendix B for further details.

[4]The HL estimator is a robust estimator calculated as the median of all Walsh averages, i.e., the pairwise means formed from every self-pair and unique unordered pair. To estimate its CIs we use a percentile bootstrap with 1000 iterations.

[5]All reported EDs have been normalised by the maximal sentence length: $ED(s_1, s_2) = ED_{raw}(s_1, s_2)/max(len(s_1), len(s_2)))$ with s_1 and s_2 being two sentences, $len(s)$ the word count of sentence s, and ED_{raw} the non-normalised ED.

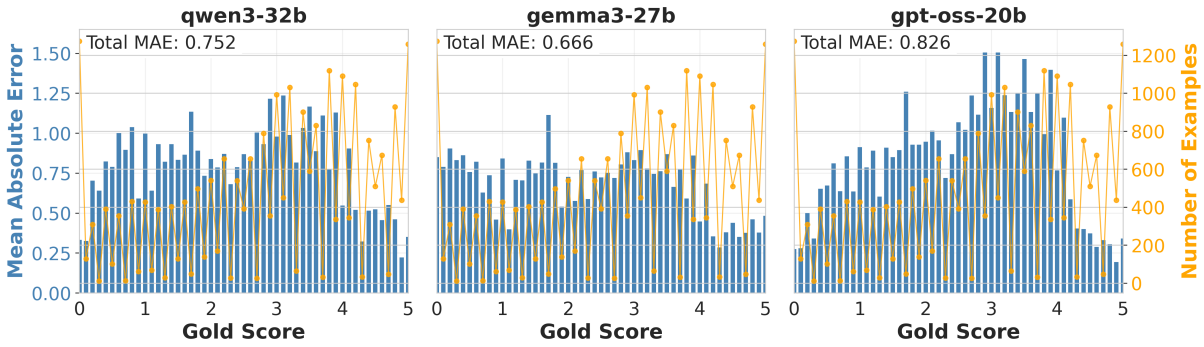

Figure 3: LLM judge mean average error (MAE) and number of examples per gold score on MTEB STS.

3 Experiments

In this section, we present the results of our experiments. We begin by outlining the implementation details, followed by the results of the LLM judge evaluation. We then examine how the selected LLM judge rates various generative models for paraphrasing and use the best paraphraser for PTEB to evaluate various embedding models across STS and non-STS tasks as well as non-English languages. Finally, we validate PTEB using human evaluation and a prompt sensitivity analysis.

3.1 Implementation

For the LLM judge and paraphraser selection, we assessed 3 generative open-weights models: `gemma 3-27b` (Gemma Team et al., 2025), `gpt-oss-20b` (OpenAI et al., 2025), and `qwen3-32b` (Yang et al., 2025). Our selection criteria were: (1) open weights to enable reproducibility and auditability; (2) strong general reasoning capabilities based on public benchmarks; (3) moderate size (20-32B parameters) to run on consumer hardware with quantisation; and (4) availability via widely-used inference frameworks (Ollama, vLLM). These constraints ensure that a wide range of researchers and practitioners can reproduce and extend PTEB. Implementation details including hyperparameters can be found in Appendix C. As encoders, we evaluated `qwen3-8b` (Yang et al., 2025), `mxbai-embed-large-v1` (Lee et al., 2024), `e5-mistral-7b-instruct` (Wang et al., 2024a,b), and `embeddinggemma-300m` (Vera et al., 2025) as SOTA encoders. As a baseline, we included the Sentence Transformer `all-mpnet-base-v2` (Reimers and Gurevych, 2020). These models were selected to include both models that currently lead or rank high on the MTEB leaderboard and a baseline model with lower performance. To ensure consistency between original and paraphrased conditions, the evaluation pipeline was re-implemented following the MTEB specification in Muennighoff et al. (2023).[6] For hyperparameters see Appendix C; for all prompts Appendix D.

Dataset	gemma3-27b	gpt-oss-20b	qwen3-32b
BIOSSES	**88.53**	83.75	80.94
SICK-R	**76.61**	70.84	72.21
STS12	**75.13**	66.95	70.42
STS13	**89.12**	87.07	87.60
STS14	**83.82**	81.95	81.55
STS15	**89.87**	87.26	86.70
STS17	**91.42**	88.99	87.53
STS22	66.87	68.82	**70.59**
STSB	**87.26**	84.50	85.67
Average	**83.18**	80.02	80.89

Table 2: Spearman's rank correlation between LLM Judge similarity scores and gold ratings for STS datasets; best scores bold. (in %)

3.2 Experimental Results

3.2.1 LLM Judge Evaluation

Analysing the mean absolute error (MAE) relative to the gold rating distribution (that is, the similarity scores) shows that all models have lower errors on sentence pairs with high (≥ 4) similarity scores (Figure 3). However, while Qwen3 and GPT-OSS also have lower MAEs on sentence pairs with labels close to 0, this does not hold for the Gemma-3 model. Table 2 presents the results for each LLM judge per dataset. Overall, `gemma3-27b` achieved the highest performance on MTEB STS. With the exception of STS22 the model outperformed GPT-

[6] Our scripts are not identical with the current MTEB-evaluation but follow the original paper, e.g. MTEB now undersamples classification data to include 8 examples per label instead of using the complete train/test splits.

OSS and Qwen3 on all datasets. Examining performance by dataset and various word count bins (averaged over all 3 LLM judges), Figure 4 shows that sentences with $\geq$ 20 words yield the lowest performance (average score 64.49%), while sentences with $<$ 10 words achieve an average Spearman's rank correlation of $\geq$ 80%. In particular, on long sentence pairs in STS14, STS12, and STSB the LLM judges score only 31.59%, 52.99% and 54.26%.

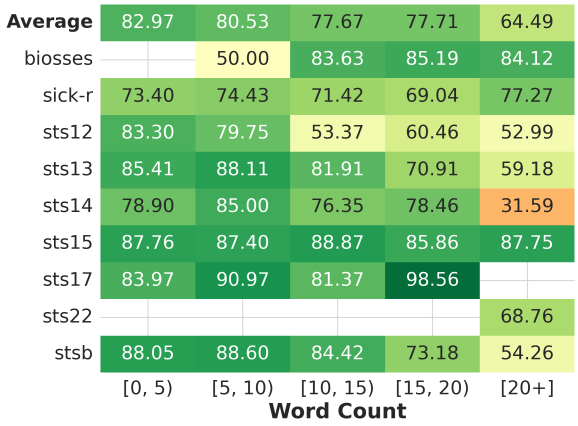

Figure 4: Spearman's rank correlation between LLM judge similarity scores and gold ratings for different word count bins; empty cells indicate that the dataset does not contain sentences with that word count. (in %)

To assess the effect of model size, we performed an ablation by varying the number of parameters (Table 3). Gemma-3 12B performs comparably to the 27B model (82.76% vs. 83.18%). By contrast, the 4B and, especially, the 1B and 270M variants exhibit substantial performance drops (72.74%, 33.28%, and 1.9% respectively).

STS Judge Model Size				
270M	**1B**	**4B**	**12B**	**27B**
1.9%	33.28%	72.74%	82.76%	83.18%

Table 3: LLM judge ablation (parameter count) for Gemma-3 grading semantic similarity on STS datasets.

3.2.2 Paraphrase Model Evaluation

Using `gemma3-27b` as the LLM judge, we observe that also as a paraphraser it not only achieves semantic similarity scores close to the model with the highest similarity score, `gpt-oss-20b`, but produces paraphrases with larger ED at a lower runtime (Table 4). The larger ED is illustrated by the examples shown in Figure 5. Hence, overall, `gemma3-27b` is the most suitable paraphraser for PTEB.

To mitigate potential within-model bias arising from using `gemma3-27b` as both the paraphraser and the LLM judge, we validated the semantic similarity ratings using `gpt-oss-20b` and `qwen3-32b` as LLM judges (Table 5). Additionally, we added `mistral-small3.2:24b` as a held-out judge. All non-Gemma LLM judges confirm high similarity scores for Gemma's paraphrases.

Paraphrase Model	**Semantic Similarity**	**Edit Distance**	**Runtime**
gemma3-27b	4.26	**0.81**	**0.27**
gpt-oss-20b	**4.44**	0.55	1.24
qwen3-32b	4.39	0.63	5.35

Table 4: Paraphrase evaluation results showing average semantic similarity (0–5 scale) and normalized edit distance compared to originals (in %), as well as runtime (seconds per text) on STS datasets; best results bold.

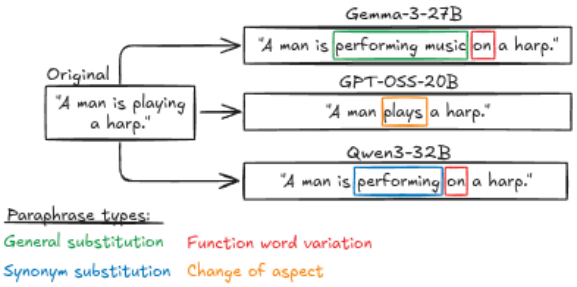

Figure 5: Paraphrase examples (types based on Bhagat and Hovy (2013)).

LLM Judge	**Average**	**HL Estimator [CI]**
gemma3-27b	4.26	4.24 [4.22, 4.30]
gpt-oss-20b	4.68	4.68 [4.65, 4.72]
qwen3-32b	4.52	4.52 [4.50, 4.54]
mistral-small3.2-24b	4.22	4.22 [4.17, 4.27]

Table 5: Semantic similarity ratings (average and Hodges-Lehmann (HL) Estimator with 95% confidence intervals) between original texts and paraphrases generated with `gemma-27b` using different LLMs as judges.

3.2.3 PTEB for STS

On average, all models decline on PTEB vs. the original STS datasets with the smallest drop for `embeddinggemma-300m` (Table 6). `e5-mistral-7b-instruct` decreases by 2.70%, while others drop by up to 5.09%. The gap is largest on STS12 (-10.47) while STS22 shows a small average improvement of +0.53. Also the

Dataset	all-mpnet-base-v2		embedding gemma-300m		mxbai-embed-large-v1		e5-mistral-7b-instruct		qwen3-embedding-8b		Average
	Orig.	PTEB	Orig.	PTEB	Orig.	PTEB	Orig.	PTEB	Orig.	PTEB	
BIOSSES	80.39	77.22	70.27	64.67	86.09	**84.06**	84.61	77.68	**86.29**	81.44	
		Δ-3.17 / ±2.23		Δ-5.60 / ±1.61		Δ-2.03 / ±1.11		Δ-6.93 / ±1.01		Δ-4.85 / ±1.53	Δ-4.52
SICK-R	80.60	77.45	71.99	70.28	82.79	78.24	80.76	79.01	**84.97**	**81.88**	
		Δ-3.15 / ±0.09		Δ-1.71 / ±0.07		Δ-4.55 / ±0.07		Δ-1.75 / ±0.05		Δ-3.09 / ±0.07	Δ-2.85
STS12	72.64	61.85	63.53	54.95	79.07	67.91	75.83	**68.23**	**81.41**	67.21	
		Δ-10.79 / ±0.29		Δ-8.58 / ±0.41		Δ-11.16 / ±0.24		Δ-7.60 / ±0.32		Δ-14.20 / ±0.30	Δ-10.47
STS13	83.49	78.91	67.71	73.70	**89.81**	84.19	84.31	**84.35**	87.89	82.90	
		Δ-4.58 / ±0.56		Δ+5.99 / ±0.64		Δ-5.62 / ±0.33		Δ+0.04 / ±0.35		Δ-4.99 / ±0.41	Δ-1.83
STS14	78.00	72.14	64.95	65.03	**85.22**	**78.41**	79.26	78.12	83.32	77.10	
		Δ-5.86 / ±0.26		Δ+0.08 / ±0.19		Δ-6.81 / ±0.12		Δ-1.14 / ±0.21		Δ-6.22 / ±0.22	Δ-3.99
STS15	85.66	81.58	77.19	74.47	89.34	85.05	85.20	86.20	**89.56**	**86.50**	
		Δ-4.08 / ±0.11		Δ-2.72 / ±0.19		Δ-4.29 / ±0.20		Δ+1.00 / ±0.10		Δ-3.06 / ±0.12	Δ-2.63
STS17	90.61	85.06	82.61	75.28	89.22	85.00	88.86	85.61	**92.19**	**88.60**	
		Δ-5.55 / ±0.25		Δ-7.33 / ±0.70		Δ-4.22 / ±0.28		Δ-3.25 / ±0.31		Δ-3.59 / ±0.21	Δ-4.79
STS22	68.31	69.39	58.05	63.21	69.02	68.97	69.66	67.49	**69.98**	68.61	
		Δ+1.08 / ±0.48		Δ+5.16 / ±0.67		Δ-0.05 / ±0.38		Δ-2.17 / ±0.74		Δ-1.37 / ±0.66	Δ+0.53
STSB	83.43	74.90	67.45	66.97	**89.29**	82.20	84.58	82.02	88.50	**84.56**	
		Δ-8.53 / ±0.35		Δ-0.48 / ±0.53		Δ-7.09 / ±0.36		Δ-2.56 / ±0.38		Δ-3.94 / ±0.32	Δ-4.52
Average	80.35	75.39	69.31	67.62	84.43	79.34	81.45	78.75	**84.90**	**79.87**	
		Δ-4.96		Δ-1.69		Δ-5.09		Δ-2.70		Δ-5.03	Δ-3.90

Table 6: Embedding model performance on original and PTEB STS datasets. Δ denotes the difference with Red indicating performance decreases and Green improvements. Sample standard deviation over the different paraphrases denoted by ±. Bold marks row-wise best for original and paraphrased text separately. ($n = 6$, in %)

all-mpnet-base-v2 vs.	Δ_{HL} [CI]	p_{Holm}
embedding-gemma-300m	+7.33 [7.02, 8.00]	6.50e-10
mxbai-embed-large-v1	-3.74 [-5.00, -3.14]	1.33e-08
e5-mistral-7b-instruct	-3.48 [-4.37, -2.56]	1.40e-07
qwen3-embedding-8b	-4.53 [-4.82, -4.19]	1.48e-09

Table 7: Comparison of encoders on PTEB STS vs. `all-mpnet-base-v2` as baseline. Δ_{HL} is the Hodges-Lehmann estimator of the location shift with 95% CI; negative values favour the listed model. P-values are Holm-adjusted. Based on 9×6 dataset-run pairs.

	Embedding Model Size		
	0.6B	**4B**	**8B**
Original STS Score	81.42	84.26	**84.90**
PTEB STS Score	77.19	78.72	**79.87**
Runtime [s]	**135**	201	263
VRAM [GB]	**1.1**	7.6	14.1

Table 8: Qwen3 embedding ablation on the numbers of parameters for STS; STS scores in %. Runtime is the time for encoding only. VRAM refers to memory usage of the model only. Best in bold. ($n = 6$ runs)

standard deviation across runs differs substantially between datasets. On SICK-R and STS15 the sample standard deviation (σ) is at the lower end ($\sigma \leq 0.09$ and $\sigma \leq 0.20$ respectively) in contrast to BIOSSES ($\sigma \in [1.01, 2.23]$).

To test statistical significance, we use the Wilcoxon signed-rank test, which is widely applicable in NLP (Dror et al., 2018).[7] Pooling over the 9×6 STS dataset-run pairs, the negative HL location shifts (Δ_{HL}) and Holm-corrected (Holm, 1979)[8] p-values $\ll \alpha = 0.05$ show that all embedding models except `embeddinggemma-300m` outperform the baseline significantly (see Table 7; for a more conservative analysis averaging across runs before testing, see Table 14 in Appendix E).

[7]Following concerns that bootstrapping requires larger sample sizes (Søgaard et al., 2014; Hesterberg, 2015), we avoid it for significance testing.

[8]The Holm-method adjusts p-values when testing multiple hypotheses. It sorts the m hypotheses by their p-values (p_i) and then checks, starting with the lowest p-value, for a given significance level α if $\forall i \in \{1, \ldots, m\} : p_i < \frac{\alpha}{m-i+1}$ (Calonico and Galiani, 2025)).

Task	all-mpnet-base-v2		embedding gemma-300m		mxbai-embed-large-v1		e5-mistral-7b-instruct		qwen3-embedding-8b		Average
	Orig.	PTEB	Orig.	PTEB	Orig.	PTEB	Orig.	PTEB	Orig.	PTEB	
STS 9 datasets (see Table 6)	80.35	75.39	69.31	67.62	84.43	79.34	81.45	78.75	**84.90**	**79.87**	
	Δ-4.96 / ±1.37		Δ-1.69 / ±0.49		Δ-5.09 / ±0.40		Δ-2.70 / ±0.43		Δ-5.03 / ±0.42		Δ-3.89
Classification Banking77	92.23	85.90	85.83	81.16	**93.66**	**87.86**	90.99	84.90	92.85	86.77	
	Δ -6.33 / ±0.48		Δ -4.67 / ±0.37		Δ -5.80 / ±0.39		Δ -6.09 / ±0.43		Δ -6.08 / ±0.37		Δ-5.79
Clustering TwentyNewsGr.	50.18	47.44	32.99	34.51	51.25	48.52	45.61	39.57	**55.51**	**50.03**	
	Δ -2.74 / ±0.71		Δ +1.52 / ±0.51		Δ -2.73 / ±0.56		Δ -6.04 / ±0.99		Δ -5.48 / ±0.75		Δ-3.09
Pair Classif. TwitterSemEval	73.85	71.65	59.03	63.90	**78.54**	73.65	75.36	74.40	72.22	**74.21**	
	Δ -2.20 / ±0.39		Δ +4.87 / ±0.10		Δ -4.89 / ±0.37		Δ -0.96 / ±0.49		Δ +1.99 / ±0.45		Δ-0.64
Reranking AskUbuntuDupQ.	65.81	64.30	48.53	48.86	65.20	63.78	59.96	59.31	**67.55**	**64.98**	
	Δ -1.51 / ±0.40		Δ +0.33 / ±0.53		Δ -1.42 / ±0.30		Δ -0.65 / ±0.46		Δ -2.57 / ±0.26		Δ-1.16
Retrieval ArguAna	45.82	45.38	29.65	33.05	65.11	61.36	53.29	49.54	**75.26**	**73.30**	
	Δ -0.44 / ±0.28		Δ +3.40 / ±0.28		Δ -3.75 / ±0.20		Δ -3.75 / ±0.38		Δ -1.96 / ±0.23		Δ-1.30
Summarisation SummEval	26.25	23.55	20.35	15.57	35.55	33.02	34.24	32.27	**38.87**	**32.95**	
	Δ -2.70 / ±0.65		Δ -4.78 / ±0.58		Δ -2.53 / ±0.46		Δ -1.97 / ±1.15		Δ -5.92 / ±1.27		Δ-3.58
Average	62.07	59.09	49.38	49.24	67.68	63.93	62.99	59.82	**69.59**	**66.02**	
	Δ -2.98		Δ -0.15		Δ -3.74		Δ -3.17		Δ -3.58		Δ-2.72

Table 9: Embedding model performance across STS and Non-STS tasks (English); the STS total (from Table 6) is included as the first row. Δ denotes differences, Red marks decreases and green improvements. ± denotes the standard deviation on PTEB. Bold marks row-wise best for Original and PTEB separately. ($n = 6$ runs; in %)

3.2.4 Ablation on STS Embedding Model Size

To analyse the influence of the size of the encoder, we conducted an ablation study using the Qwen3 models (Table 8). The performance gap between the original and paraphrased datasets remains similar across all 3 Qwen3 variants, with the smallest model even displaying the narrowest gap.

3.2.5 PTEB for Non-STS Tasks

We extend PTEB to 6 additional MTEB tasks in order to incorporate a broad range of tasks ranging from straightforward binary decisions to nuanced clustering of sentences. Table 9 shows that the average performance of all models drops on PTEB compared to original texts, although particularly `embeddinggemma-300m` improves on multiple tasks. `qwen3-8b` is the best-performing model in 5 of the 7 tasks on original data and 6 of 7 on PTEB. (For an additional visualisation see Appendix F.)

3.2.6 Multilingual PTEB

We validate our results on 6 datasets from 6 different tasks that include non-English data (incl. the multilingual version of STS17) covering 25 languages. We examine both widely-studied languages (e.g. Arabic) and under-resourced ones (e.g. Danish) incl. 13 from sub-Saharan Africa (e.g. Suwaheli, Hausa, Amharic) to evaluate performance across linguistic boundaries (see Table 10 for results; see Appendix A for the languages included).[9] We used the same encoders as before if they are natively multilingual, or their multilingual variants alternatively. Consistent with English datasets, we find the largest drops on the classification task followed by STS, and clustering. An exception is the Russian RubQ dataset on which all models improve. As expected, for each task the average decreases are more pronounced on the English datasets compared to non-English datasets.

3.2.7 Human Evaluation

Languages for human evaluation were selected to (1) cover high and low resource languages (including low-resource Sub-Saharan languages like Hausa and Swahili), (2) represent a wide range of PTEB tasks (spanning Classification, Clustering, Pair Classification, Reranking, and STS), and (3) encompass diverse language families (e.g., Afroasiatic, Indo-European, Niger-Congo, and Turkic). The human similarity ratings ranged from 3.34 (Hausa) to 4.43 (Arabic), with an average of 4.05 and a pooled median of 4.00 (Table 11). This confirms a high degree of semantic similarity between the generated paraphrases and the original text. All languages exceeded the mean acceptability threshold. While the majority of sentence pairs were rated

[9] We did not include a non-English dataset for the summarisation task, as MTEB does not provide one.

Task	paraphrase-multilingual-mpnet-base-v2		embedding gemma-300m		multilingual-e5-large-instruct		qwen3-embedding-8b		Average
	Orig.	PTEB	Orig.	PTEB	Orig.	PTEB	Orig.	PTEB	
Classification AmazonCounterf.	88.54	85.55 Δ -2.99 / ±0.42	**90.51**	86.11 Δ -4.40 / ±0.36	89.08	86.32 Δ -2.77 / ±0.17	89.83	**87.04** Δ -2.79 / ±0.31	Δ-3.24
Clustering MasakhaNEWS	18.98	19.41 Δ +0.43 / ±0.87	16.17	15.51 Δ -0.67 / ±1.43	24.77	21.04 Δ -3.72 / ±2.31	**25.49**	**24.42** Δ -1.08 / ±1.53	Δ-1.26
Pair Classification RTE3	**89.14**	**89.29** Δ +0.15 / ±0.05	86.95	86.49 Δ -0.46 / ±0.24	87.62	87.23 Δ -0.39 / ±0.13	87.35	87.24 Δ -0.11 / ±0.18	Δ-0.20
Reranking RuBQReranking	58.74	58.93 Δ +0.19 / ±0.68	22.62	24.11 Δ +1.49 / ±0.60	69.51	69.80 Δ +0.29 / ±0.13	**75.07**	**75.08** Δ +0.01 / ±0.30	Δ+0.50
Retrieval TwitterHjerne	58.26	57.78 Δ -0.48 / ±1.21	29.15	25.70 Δ -3.45 / ±0.63	72.62	73.16 Δ +0.54 / ±1.18	**78.72**	**80.54** Δ +1.82 / ±0.89	Δ-0.40
STS STS17	**83.46**	79.98 Δ -3.48 / ±0.19	61.89	59.74 Δ -2.15 / ±0.51	72.82	70.86 Δ -1.96 / ±0.26	83.35	**80.24** Δ -3.11 / ±0.20	Δ-2.67
Average	66.19	65.16 Δ -1.03	51.22	49.61 Δ -1.61	69.40	68.07 Δ -1.33	**73.30**	**72.43** Δ -0.88	Δ-1.21

Table 10: Multilingial embedding model performance on original and PTEB datasets (25 languages); Δ denotes differences, Red marks decreases and green improvements. ± denotes the standard deviation on PTEB. Bold marks row-wise best for Original and PTEB separately. ($n = 6$ runs; in %)

Language	No. of Ratings	Mean Similarity	Median Similarity
Arabic (ARA)	2x50	4.43	5.00
French (FRA)	2x50	3.78	4.00
German (DEU)	2x50	4.40	5.00
Hausa (HAU)	2x50	3.34	4.00
Russian (RUS)	2x50	4.16	5.00
Spanish (SPA)	2x50	4.01	4.00
Swahili (SWA)	2x50	3.91	4.00
Turkish (TUR)	2x50	4.40	5.00
Total	**2x400**	**4.05**	**4.00**

Table 11: Human evaluation of the similarity between original texts and paraphrases (on scale from 0 to 5).

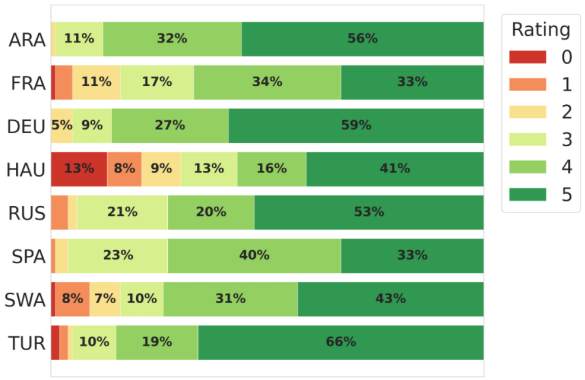

Figure 6: Human paraphrase similarity rating distribution (based on $n = 2 \times 50$ ratings per language).

3.00 or higher, 13% of sentence pairs in Hausa received a similarity score of 0.00 (Figure 6).

3.2.8 STS Paraphrase Prompting Sensitivity

Since LLMs can be sensitive to prompt changes (Mizrahi et al., 2024), we tested the influence of 3 variations of the paraphrase prompt and compared the PTEB STS scores and the ED between original and paraphrased texts (Table 12). We find limited differences ($\leq 0.92\%$ for the score and ≤ 0.03 for the ED).

Paraphrase Prompt	PTEB STS Score	Edit Distance
Default	74.92	0.81
Variant 1	74.24	0.82
Variant 2	74.82	0.82
Variant 3	75.16	0.84

Table 12: PTEB STS scores (English; in %) and the corresponding edit distances to the original text for the default paraphrase prompt and 3 differently phrased prompt variants; see Appendix D.4 for the prompts. (Based on all-mpnet-base-v2 embeddings; $n = 1$ runs)

4 Discussion

Our method demonstrates that the 3 generative LLMs evaluated in our study are well-suited to rate semantic similarity with average STS scores $> 80\%$ (Table 2). Gemma-3-27b even outperformed 3 of the 5 embedding models on average on the original STS data (see Table 2 vs. Table 6).[10]

[10]With the limitation that Gemma-3-27b was only evaluated on one run and this is, hence, not a like-for-like comparison.

Furthermore, the evaluated LLMs generate paraphrases with high semantic similarity while introducing token-level variation from the original text (Table 4). We found that `gemma3-27b` generated more textually diverse paraphrases at higher throughput: average ED 0.81 vs. 0.55 and 0.63 at a ~5× to ~20× speed-up compared to `gpt-oss-20b` and `qwen3-32b` respectively.

Based on our human evaluation for 8 languages, we found that the generated paraphrases are highly similar to the original sentences (mean: 4.05; median: 4.00; see Table 11). However, for Hausa, Swahili, and French, only 70%, 84%, and 84% of sentence pairs met the similarity threshold of 3.00. While this may indicate limited paraphrasing capabilities for low-resource languages like Hausa and Swahili, it is surprising that a high-resource language like French exhibits similar issues.

Across the 25 languages, we observe that all models exhibit lower average scores on PTEB. This drop is most pronounced in English (-2.72pt vs. -1.21pt; Table 9 vs. Table 10), suggesting that encoders (particularly high-performing ones) may be overfitted to MTEB, relying on shortcuts in the token space rather than semantic understanding. Classification tasks show the largest average performance declines across both English and non-English datasets. We hypothesise that this occurs because classification relies heavily on specific keyword triggers that are disrupted by paraphrasing.

Except for STS22 and RubQ, 18 out of the 20 datasets show lower PTEB scores averaged over all embedding models vs. original scores. For STS22, this can largely be attributed to EmbeddingGemma while the relatively small performance drops for other encoders may be explained by STS22's higher word count (461 vs. 61 for other STS datasets; see Table 1). Qualitative observations suggest that the longer texts in STS22 are more robust to lexical changes, whereas paraphrases of shorter sentences rely heavily on synonym replacement. In these shorter contexts, individual tokens carry disproportionate weight, potentially distorting the embeddings. For RubQ we believe this is partly due to a higher robustness of the reranking task as also observed for the English datasets (Table 9, Table 10). Moreover, RubQ's candidate documents have a substantially higher average word count (63 words vs. 9 for AskUbuntuDuplicateQuestions).

Furthermore, English datasets with initial scores $\geq$ 80 (e.g. BIOSSES, SICK-R, STS13/15/17/B, and Banking77) showed relatively large performance gaps (many models -3 to -6pts; also see Figure 7 in the Appendix). This suggests that the initially high scores may depend on shortcuts in token space which are neutralised by paraphrasing. It shows that PTEB's new evaluation method adds a stress test for semantic invariance, mitigating concerns that MTEB may favour lexical shortcuts and overfitted embedding models. However, neither do we claim that any specific model is more overfitted than another one nor do we make general claims on the performance of individual models as the aim of our study is to present a novel evaluation method; not a comprehensive benchmarking study.

Unexpectedly, smaller encoders were not more sensitive to paraphrasing to a certain extend: Our ablation study on the Qwen3 encoder size showed the similar performance for the 4B vs. 8B parameter models (Table 8). For the datasets in our study, this indicates that robust performance under paraphrase stress can be achieved with smaller embedding models. Furthermore, the prompt sensitivity analysis showed that PTEB is robust to prompt changes further mitigating potential concerns that the results might be brittle due to the use of generative LLMs (Table 12).

5 Conclusion

We introduce PTEB, a sentence embedding benchmark that stochastically paraphrases MTEB data at evaluation time. Using LLM judges and human evaluation, we show that the LLMs in scope yield semantics-preserving yet token-diverse paraphrases. Building on this, we demonstrate that the average performance of the tested embedding models drops across 7 NLP tasks when paraphrased, showing that PTEB effectively uncovers generalisation failures. For non-English languages, we included 6 multilingual datasets covering 25 languages incl. low resource (e.g., African) languages. Ablations over LLM judge and sentence embedding model sizes show runtime–quality trade-offs for the models in scope and that smaller models (12B parameters for LLM judges and 4B for embedding models) perform comparably to larger ones in our study. Our experiments confirm that PTEB yields statistically robust results across multiple runs with different random seeds and even if the paraphrase prompt is varied. In general, PTEB is a novel, more robust evaluation methodology well suited to augment current embedding benchmarks.

6 Limitations

LLM judge and paraphrase model evaluation. The evaluations of the LLM judge and paraphrase models were limited to a single run. This was sufficient for our goal of demonstrating an end-to-end methodology for PTEB that grounds generative model selection in human gold ratings without human evaluation. For production scenarios, our method can serve as a template and be expanded accordingly. Alternatively, one might pragmatically select a paraphrase model directly.

Human validation. The human evaluation in our work is limited to 8 languages and 400 sentence pairs (each rated by 2 evaluators). For paraphrasing of English datasets we relied on an automated approach using LLM judges which were only indirectly validated by grounding them in STS gold labels but not validated through human evaluation.

PTEB paraphrase model. Our embedding model evaluation relies exclusively on `gemma3-27b` paraphrases. While the cross-model-family validation (Table 5) confirmed consistent semantic similarity scores across different LLM judges, future work should examine whether the paraphrasing style itself affects embedding model rankings by running PTEB with different paraphrasers.

In an initial analysis using the STSB dataset, we compared the edit distance between paraphrases and original texts to the performance drops on PTEB for `gemma3-27b`, `gpt-oss-20b`, and `qwen3-32b` (see Figure 8 in Appendix F). We found that generative models that generate paraphrases with larger edit distances are associated with larger performance drops of embedding models on PTEB. That is, in our initial analysis 4 out of 5 embedding models show the largest performance drops when using `gemma3-27b` as a paraphrase model (which is the paraphraser generating the largest EDs; see Table 4). Similarly, `qwen3-32b`, which produces the lowest EDs, yields the smallest performance drops for 4 out of 5 embedding models. Despite the need for further analysis, we interpret this as initial evidence that using the ED as a selection criterion is essential, since our goal was to generate paraphrases with substantial token-level changes. Which was exactly one of the reasons to choose `gemma3-27b` (see Table 4) as also illustrated in a non-cherry picked example in Figure 5.

7 Ethical Considerations

Model bias. However, LLMs (and, hence, PTEB) may replicate or amplify biases present in their training data (Bender et al., 2021). To support ethical scrutiny and reproducibility, we use only open-weight models (i.e., no commercial APIs), enabling researchers to reproduce results, pin versions, and audit behaviour including limited inspection of internals without API accessibility limitations or usage restrictions. While it does not by itself resolve bias, it improves assessability for ethical research.

Personal data. No personal data was collected and all evaluations are based on public datasets.

Crowd worker compensation. We ensured appropriate and equitable compensation by paying all annotators an appropriate and identical rate regardless of their geographic location. At $0.25 per sentence pair, it corresponds to an estimated hourly wage of $36.

Overall, we did not identify major ethical concerns arising from this work.

Acknowledgments

We thank the authors of the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2023) and the Massive Multilingual Text Embedding Benchmark (MMTEB) (Enevoldsen et al., 2025) for making their code and the included datasets publicly available. While PTEB builds on MTEB and MMTEB, it is an independent extension and is not affiliated with or endorsed by the MTEB or MMTEB authors.

We also thank the anonymous reviewers and human evaluators for their valuable contributions.

We acknowledge the use of AI, such as Anthropic's Claude Code, OpenAI's ChatGPT, and Google's Gemini for assisted coding and writing, e.g., for improving the language of our paper.

This research was partially supported by the Horizon Europe project GenDAI (Grant Agreement ID: 101182801) and by the ADAPT Research Centre at Munster Technological University. ADAPT is funded by Taighde Éireann – Research Ireland through the Research Centres Programme and co-funded under the European Regional Development Fund (ERDF) via Grant 13/RC/2106_P2.

References

David Ifeoluwa Adelani, Hannah Liu, Xiaoyu Shen, Nikita Vassilyev, Jesujoba O. Alabi, Yanke Mao, Haonan Gao, and En-Shiun Annie Lee. 2024. SIB-200: A Simple, Inclusive, and Big Evaluation Dataset for Topic Classification in 200+ Languages and Dialects. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 226–245, St. Julian's, Malta. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 Task 10: Multilingual Semantic Textual Similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A pilot on semantic textual similarity. In **SEM 2012 - 1st Joint Conference on Lexical and Computational Semantics*, volume 2, pages 385–393.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic Textual Similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pages 610–623, New York, NY, USA. Association for Computing Machinery.

Rahul Bhagat and Eduard Hovy. 2013. Squibs: What Is a Paraphrase? *Computational Linguistics*, 39(3):463–472. Place: Cambridge, MA Publisher: MIT Press.

Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A Full-Text Learning to Rank Dataset for Medical Information Retrieval. In Nicola Ferro, Fabio Crestani, Marie-Francine Moens, Josiane Mothe, Fabrizio Silvestri, Giorgio Maria Di Nunzio, Claudia Hauff, and Gianmaria Silvello, editors, *Advances in Information Retrieval*, volume 9626, pages 716–722. Springer International Publishing, Cham.

Sebastian Calonico and Sebastian Galiani. 2025. Beyond Bonferroni: Hierarchical Multiple Testing in Empirical Research. *Preprint*, arXiv:2507.19610.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient Intent Detection with Dual Sentence Encoders. *Preprint*, arXiv:2003.04807.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Xi Chen, Ali Zeynali, Chico Camargo, Fabian Flöck, Devin Gaffney, Przemyslaw Grabowicz, Scott Hale, David Jurgens, and Mattia Samory. 2022. SemEval-2022 Task 8: Multilingual news article similarity. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1094–1106, Seattle, United States. Association for Computational Linguistics.

Isaac Chung, Imene Kerboua, Marton Kardos, Roman Solomatin, and Kenneth Enevoldsen. 2025. Maintaining MTEB: Towards Long Term Usability and Reproducibility of Embedding Benchmarks. *Preprint*, arXiv:2506.21182.

Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. 2018. How Many Random Seeds? Statistical Power Analysis in Deep Reinforcement Learning Experiments. *Preprint*, arXiv:1806.08295.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Zihao Wu, Lin Zhao, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2023. ChatAug: Leveraging ChatGPT for Text Data Augmentation.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Rotem Dror, Segev Shlomov, and Roi Reichart. 2019. Deep Dominance - How to Properly Compare Deep Neural Models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2785, Florence, Italy. Association for Computational Linguistics.

Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński,

Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Rystrøm, Roman Solomatin, and 67 others. 2025. MMTEB: Massive Multilingual Text Embedding Benchmark. *Preprint*, arXiv:2502.13595.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating Summarization Evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Manuel Frank and Haithem Afli. 2025. GASE: Generatively augmented sentence encoding. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5444–5461, Suzhou, China. Association for Computational Linguistics.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 Technical Report. *Preprint*, arXiv:2503.19786.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics.

Samarth Goel, Reagan J. Lee, and Kannan Ramchandran. 2025. SAGE: A Realistic Benchmark for Semantic Understanding. *Preprint*, arXiv:2509.21310.

Tim C. Hesterberg. 2015. What Teachers Should Know About the Bootstrap: Resampling in the Undergraduate Statistics Curriculum. *The American Statistician*, 69(4):371–386.

J. L. Hodges and E. L. Lehmann. 1963. Estimates of Location Based on Rank Tests. *The Annals of Mathematical Statistics*, 34(2):598–611.

Soren Vejlgaard Holm. 2024. Are gllms danoliterate? Benchmarking generative NLP in danish. Accessed: 2025-11-23.

Sture Holm. 1979. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics*, 6(2):65–70.

Ken Lang. 1995. NewsWeeder: Learning to Filter Netnews. In *Machine Learning Proceedings 1995*, pages 331–339, San Francisco (CA). Morgan Kaufmann.

Sean Lee, Aamir Shakir, Darius Koenig, and Julius Lipp. 2024. Open source strikes bread - new fluffy embeddings model. Accessed: 2025-05-10.

Hongji Li, Andrianos Michail, Reto Gubelmann, Simon Clematide, and Juri Opitz. 2025. Sentence smith: Controllable edits for evaluating text embeddings. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26439–26456, Suzhou, China. Association for Computational Linguistics.

Zi Liang, Liantong Yu, Shiyu Zhang, Qingqing Ye, and Haibo Hu. 2025. How Much Do Large Language Model Cheat on Evaluation? Benchmarking Overestimation under the One-Time-Pad-Based Framework. *Preprint*, arXiv:2507.19219.

Edward Ma. 2019. NLP augmentation. Accessed: 2025-08-01.

M. Marelli, Stefano Menini, Marco Baroni, L. Bentivogli, R. Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *International Conference on Language Resources and Evaluation*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *Preprint*, arXiv:1301.3781.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of What Art? A Call for Multi-Prompt LLM Evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. MTEB: Massive Text Embedding Benchmark. *Preprint*, arXiv:2210.07316.

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

OpenAI, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, and 107 others. 2025. Gpt-oss-120b & gpt-oss-20b Model Card. *Preprint*, arXiv:2508.10925.

Nils Reimers and Iryna Gurevych. 2017. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2018. Why Comparing Single Performance Scores Does Not Allow to Draw Conclusions About Machine Learning Approaches. *preprint, arXiv:1803.09578*.

Nils Reimers and Iryna Gurevych. 2020. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 3982–3992.

Ivan Rybin, Vladislav Korablinov, Pavel Efimov, and Pavel Braslavski. 2021. RuBQ 2.0: An Innovated Russian Question Answering Dataset. In *The Semantic Web*, pages 532–547, Cham. Springer International Publishing.

Anders Søgaard, Anders Johannsen, Barbara Plank, Dirk Hovy, and Hector Martínez Alonso. 2014. What's in a p-value in NLP? In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 1–10, Ann Arbor, Michigan. Association for Computational Linguistics.

Gizem Soğancıoğlu, Hakime Öztürk, and Arzucan Özgür. 2017. BIOSSES: A semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics (Oxford, England)*, 33(14):i49–i58.

Raghuveer Thirukovalluru and Bhuwan Dhingra. 2025. GenEOL: Harnessing the Generative Power of LLMs for Training-Free Sentence Embeddings. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2295–2308, Albuquerque, New Mexico. Association for Computational Linguistics.

Craig Thomson, Ehud Reiter, and Anya Belz. 2024. Common Flaws in Running Human Evaluation Experiments in NLP. *Computational Linguistics*, 50(2):795–805.

Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, Daniel Cer, Alice Lisak, Min Choi, Lucas Gonzalez, Omar Sanseviero, Glenn Cameron, Ian Ballantyne, Kat Black, Kaifeng Chen, and 70 others. 2025. EmbeddingGemma: Powerful and Lightweight Text Representations. *Preprint*, arXiv:2509.20354.

Jan Philip Wahle, Terry Ruas, Frederic Kirstein, and Bela Gipp. 2022. How Large Language Models are Transforming Machine-Paraphrased Plagiarism.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021a. TSDAE: Using Transformer-based Sequential Denoising Auto-Encoder for Unsupervised Sentence Embedding Learning. *Preprint*, arXiv:2104.06979.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024a. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *Preprint*, arXiv:2212.03533.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024b. Improving Text Embeddings with Large Language Models. *Preprint*, arXiv:2401.00368.

Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui Zheng, Zexiong Pang, Qinzhuo Wu, Zhengyan Li, Chong Zhang, Ruotian Ma, Zichu Fei, Ruijian Cai, Jun Zhao, Xingwu Hu, Zhiheng Yan, and 15 others. 2021b. TextFlint: Unified Multilingual Robustness Evaluation Toolkit for Natural Language Processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 347–355, Online. Association for Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Bill Dolan. 2015. SemEval-2015 task 1: Paraphrase and semantic similarity in Twitter (PIT). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 1–11, Denver, Colorado. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 Technical Report. *Preprint*, arXiv:2505.09388.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E. Gonzalez, and Ion Stoica. 2023. Rethinking Benchmark and Contamination for Language Models with Rephrased Samples. *Preprint*, arXiv:2311.04850.

A Datasets

Table 13 shows all PTEB datasets with the respective metrics and the languages included in this study. Further details on all datasets are available in Muennighoff et al. (2023) and Enevoldsen et al. (2025), as well as at `https://huggingface.co/mteb/datasets`.

B Human Evaluation Details

For each language, we filtered the Amazon Mechanical Turk workers based on their location:

Tasks/Datasets	Languages	Metric
Classification		Accuracy
Banking77 (Casanueva et al., 2020)	eng	
AmazonCounterfactuals (O'Neill et al., 2021)	deu, eng, jpn	
Clustering		V-measure
TwentyNewsGroups (Lang, 1995)	eng	
MasakhaNEWSClusteringS2S (Adelani et al., 2024)	amh, eng, fra, hau, ibo, lin, lug, orm, pcm, run, sna, som, swa, tir, xho, yor	
Pair Classification		Average Precision
TwitterSemEval2015 (Xu et al., 2015)	eng	
RTE3 (Giampiccolo et al., 2007)	deu, eng, fra, ita	
Reranking		MAP
AskUbuntuDupQuestions (Wang et al., 2021a)	eng	
RuBQ Reranking (Rybin et al., 2021)	rus	
Retrieval (Due to corpus length only queries are paraphrased.)		nDCG@10
ArguAna (Boteva et al., 2016)	eng	
TwitterHjerne (Holm, 2024)	dan	
Summarisation		Spearman corr.
SummEval (Fabbri et al., 2021)	eng	
Semantic Textual Similarity (STS)		Spearman corr.
BIOSSES (Soğancıoğlu et al., 2017)	eng	
SICK-R (Marelli et al., 2014)	eng	
STS12 (Agirre et al., 2012)	eng	
STS13 (Agirre et al., 2013)	eng	
STS14 (Agirre et al., 2014)	eng	
STS15 (Agirre et al., 2015)	eng	
STS17 (Cer et al., 2017)	eng, ara, deu, fra, ita, nld, spa, tur	
STS22 (Chen et al., 2022)	eng	
STSBenchmark (Cer et al., 2017)	eng	

Table 13: Overview of datasets included in PTEB. For each dataset, we report the language(s) and evaluation metric; since we only used the English subset of STS22, other languages are not listed for this dataset.

- Arabic: Algeria, Egypt, Iraq, Jordan, Lebanon, Morocco, Saudi Arabia, Sudan, Syrian Arab Republic, Tunisia
- French: Belgium, Canada, France, Switzerland
- German: Austria, Germany, Switzerland
- Hausa: Benin, Cameroon, Chad, Ghana, Niger, Nigeria, Togo
- Russian: Armenia, Belarus, Brazil, Estonia, Georgia, Germany, Kazakhstan, Kyrgyzstan, Latvia, Lithuania, Russian Federation, Ukraine, United States
- Spanish: Argentina, Bolivia, Chile, Colombia, Costa Rica, Dominican Republic, Guatemala, Honduras, Mexico, Panama, Paraguay, Peru, Puerto Rico, Spain, United States
- Swahili: Burundi, Congo (DRC), Kenya, Mozambique, Rwanda, Sudan, Tanzania (United Republic of Tanzania), Uganda
- Turkish: Turkey

All country names are spelt as on the Amazon Mechanical Turk platform.

The instructions were based on the LLM prompts (see Appendix D.) Each rating/HIT was paid $0.25. Based on our assumption that a rating/HIT takes on average 25 seconds, this equals a payment of $36 per hour. This payment was independent of the worker's location. The time limit for each worker to complete a batch was set to 1 hour. For each language we ran an initial pilot with 20 sentence pairs. In the case of one language, Danish, we were not able to collect any ratings and therefore replaced it with French - while not being a low-resource language, French is a highly relevant for PTEB since 3 of the datasets in this study include French text (see Table 13).

As a basic validation of the human ratings, we included dummy sentence pairs using an almost identical sentence pair (easy positive) and a completely dissimilar sentence pair (easy negative) in each batch. For each worker, we checked if these were rated correctly within a margin of ±1 before accepting their ratings. Overall, we employed the following validation steps for all ratings based on our 0–5 rating scale:

1. Check the ratings of the two dummy sentences.[11]
2. Prompt GPT-5.1 and Claude Sonnet 4.5 with the sentence pairs and human ratings and ask both to flag any ratings which deviate by more than 2.0 from their own similarity estimate.
3. Identify any sentence pair with a similarity rating of 2.0 and below.
4. Translate all sentence pairs identified in step 1 to 3 to English using Google Translate and manually inspect the validity of the ratings.
5. Reject workers that fail the dummy sentence pairs within the pre-defined acceptability threshold or repeatedly submitted ratings that deviate by more than 2 from the similarity rating based on the manual inspection and the LLM ratings. Validate the potential candidates to be dropped using the results from step 4.

These validation steps and criteria have been defined a priori to avoid the common mistake of ad-hoc rejection of workers or ratings leading to biased results (Thomson et al., 2024). Generally, we aimed to keep rejections to a minimum.

Based on these steps, the ratings of exactly one worker for Arabic were rejected since they incorrectly rated the dummy sentence pairs beyond the acceptability threshold. Also, the ratings of the worker in question were consistently more than 2 points below that of other Arabic workers, the two LLMs estimates, and the manual translation-based approach.

Furthermore, we obtained clarification by two additional workers whose ratings were flagged during the above process while they correctly rated the dummy sentences. Following this, their ratings were ultimately approved.

C Implementation Details

Generative Models. All generative models were run using Ollama (v0.11.7):

- `qwen3:32b`
  - Quantisation: `Q4_K_M`
  - License: Apache License Version 2.0, January 2004

[11] Note that not all workers completed the dummy sentence pair ratings.

  - Link: https://www.ollama.com/library/qwen3:32b

- gpt-oss:20b
  - Quantisation: MXFP4
  - License: Apache License Version 2.0, January 2004
  - Link: https://www.ollama.com/library/gpt-oss:20b

- gemma3:27b
  - Quantisation: Q4_K_M
  - License: Gemma Terms of Use (last modified 2024-02-21)
  - Link: https://www.ollama.com/library/gemma3:27b

- mistral-small3.2:24b
  - Quantisation: Q4_K_M
  - License: Apache License Version 2.0, January 2004
  - Link: https://www.ollama.com/library/mistral-small3.2:24b

We used the default hyperparameters and environment variables (OLLAMA_FLASH_ATTENTION=1, for example). Further information on the models is available at www.ollama.com/models.

Embedding models. All embedding models were ran using the sentence-transformers package (Reimers and Gurevych, 2020) (v3.3.1), dtype torch.bfloat16 and default max_seq_len (ranging from 384 for all-mpnet-base-v2 to 32,768 for qwen3-embedding-8B). For instruction-tuned models we use the model-specific prompts if available. Please note that some models like qwen3-embedding-8b might achieve better scores with custom prompts. Further information on the models is available at https://huggingface.co/models.

For retrieval datasets only the queries were paraphrased due to the length of the corpora. Also, for task-specific hyperparameters we applied the MTEB-defaults (v1.38.33), e.g. max_iter=100 for the Logistic Regression model in the Classification task. For a description of the evaluation protocol for each task the reader may refer to Muennighoff et al. (2023). MTEB is licensed under the Apache License 2.0.

Random Seeds. We used 1337 as a random seed for all experiments. If more than one run was performed, the seed was incremented by one after each run for the paraphrase models only. An exception are the paraphrases of BIOSSES and SICK-R. For these two no paraphrase have been generated with seed=1342 but seed=1443 instead.

Hardware. All experiments were conducted on a personal computer with the following specification:

- GIGABYTE GeForce RTX 5090 GAMING OC 32G (GPU)
- AMD Ryzen 9 9950X (CPU)
- DDR5-6400 64GB (RAM)

D Prompts

The following prompts were used for generative LLMs. Curly brackets indicate placeholders that are filled in with the corresponding variables. These prompts were designed in initial experiments using dummy sentences with the goal to provide the desired output. They are not optimised systematically. The generated paraphrases were selectively reviewed by the authors (in multiple, but not all languages).

D.1 LLM Judge Prompts

Used in step 1 of our method to rate STS:

Semantic Similarity Rating Prompt

You are an expert evaluator of semantic textual similarity. Your task is to rate how similar two sentences are in meaning.

Please rate the semantic similarity between the following two sentences on a scale from 0.0 to 5.0 using one decimal place:

– 0.0-1.0: Completely different meanings
– 1.1-2.0: Slightly related but mostly different
– 2.1-3.0: Somewhat similar with some shared meaning
– 3.1-4.0: Very similar with minor differences
– 4.0-5.0: Essentially the same meaning

Only respond with a single decimal number from 0.0 to 5.0 (one decimal place). Do not provide explanations or additional text.

Sentence 1: {sentence1}
Sentence 2: {sentence2}

Similarity rating:

D.2 Paraphrase Evaluation Prompts

Used in step 2 of our method to rate the paraphrase models:

Paraphrase Rating Prompt

You are an expert evaluator of semantic textual similarity. Your task is to rate how similar two sentences are in meaning.

Please rate the semantic similarity between the following two sentences on a scale from 0.0 to 5.0 using one decimal place:
– 0.0-1.0: Completely different meanings
– 1.1-2.0: Slightly related but mostly different
– 2.1-3.0: Somewhat similar with some shared meaning
– 3.1-4.0: Very similar with minor differences
– 4.0-5.0: Essentially the same meaning

Original: {original}
Paraphrase: {paraphrase}

Only respond with a single decimal number from 0.0 to 5.0 (one decimal place). Do not provide explanations or additional text.

Similarity rating:

D.3 Paraphrase Prompts

Used in step 2 and step 3 of our method to generate paraphrases:

Default

Rephrase the following text while keeping its original meaning. Only reply with the paraphrased text and only provide a single response — no alternatives! Do not include thinking tokens, explanations or notes.

Used for multilingual datasets to maintain consistency between input and output language:

Multilingual

Rephrase the following text while keeping its original meaning. Only reply with the paraphrased text and only provide a single response — no alternatives! Do not include thinking tokens, explanations or notes. Answer in the same language as the input text. Do not translate to English or any other language.

D.4 Paraphrase Prompt Variatons for Sensitivity Analysis

Used in the prompt sensitivity analysis to test sensitivity to prompt changes. The variants were obtained by promtping GPT-5 to rephrase the default prompt:

Variation 1

Paraphrase the text below so that the meaning stays the same. Return only one rewritten version of the text. Do not add explanations, reasoning steps, or multiple options.

Variation 2

Rewrite the given text in different words but preserve its meaning. Output exactly one paraphrase. Do not include commentary, reasoning, or alternative suggestions.

Variation 3

Provide a single paraphrased version of the following text. The meaning must remain unchanged. Do not include any notes, thoughts, or more than one response.

E Run-Aggregated Statistical Analysis

Table 14 shows the results of the statistical analysis when taking the average over the $n = 6$ runs (which reduces the number of observations from $9 \times 6 = 54$ to just 9) before calculating the HL estimated location shift and applying the Wilcoxon signed-rank test. Due to the smaller number of observations we do not calculate bootstrap CIs. The run-aggregation follows Søgaard et al. (2014), who average in their experiments for 2 different NLP tasks over 3 and 10 runs respectively.

Recommended minimum sample sizes for the bootstrap method vary considerably, from 200 (Søgaard et al., 2014) to 20 (Colas et al., 2018), with

all-mpnet-base-v2 vs.	Δ_{HL}	p_{Holm}
embeddinggemma-300m	+7.17	0.02
mxbai-embed-large-v1	-3.62	0.04
e5-mistral-7b-instruct	-3.42	0.04
qwen3-embedding-8b	-4.46	0.02

Table 14: Run-aggregated comparison of encoders on PTEB STS vs. `all-mpnet-base-v2` as baseline. Δ_{HL} is the Hodges–Lehmann estimate of the location shift with its 95% confidence interval; negative values favour the listed model. P-values are exact Wilcoxon with Holm correction. (9 datasets, average over $n = 6$ runs first)

the requirement that samples be representative of the population (Colas et al., 2018; Dror et al., 2018). In our experiments, the number of observations is $9 \times 6 = 54$ when pooling over datasets and runs (see Table 7) and 9 when averaging across runs first (see Table 14). Given these modest number of observations, we use the Wilcoxon signed-rank test for significance testing.

The results show that even in this conservative analysis, the p-values remain below $\alpha = 0.05$ after Holm-correction.

F Additional Analyses

Figure 7 visualises how the performance of embeddings models drops on PTEB vs. original data per for the various task.

Figure 8 shows how paraphrase models that generate paraphrases with larger edit distance compared to the original data (see Table 4) tend to lead to larger performance drops for embedding models on the STSBenchmark dataset.

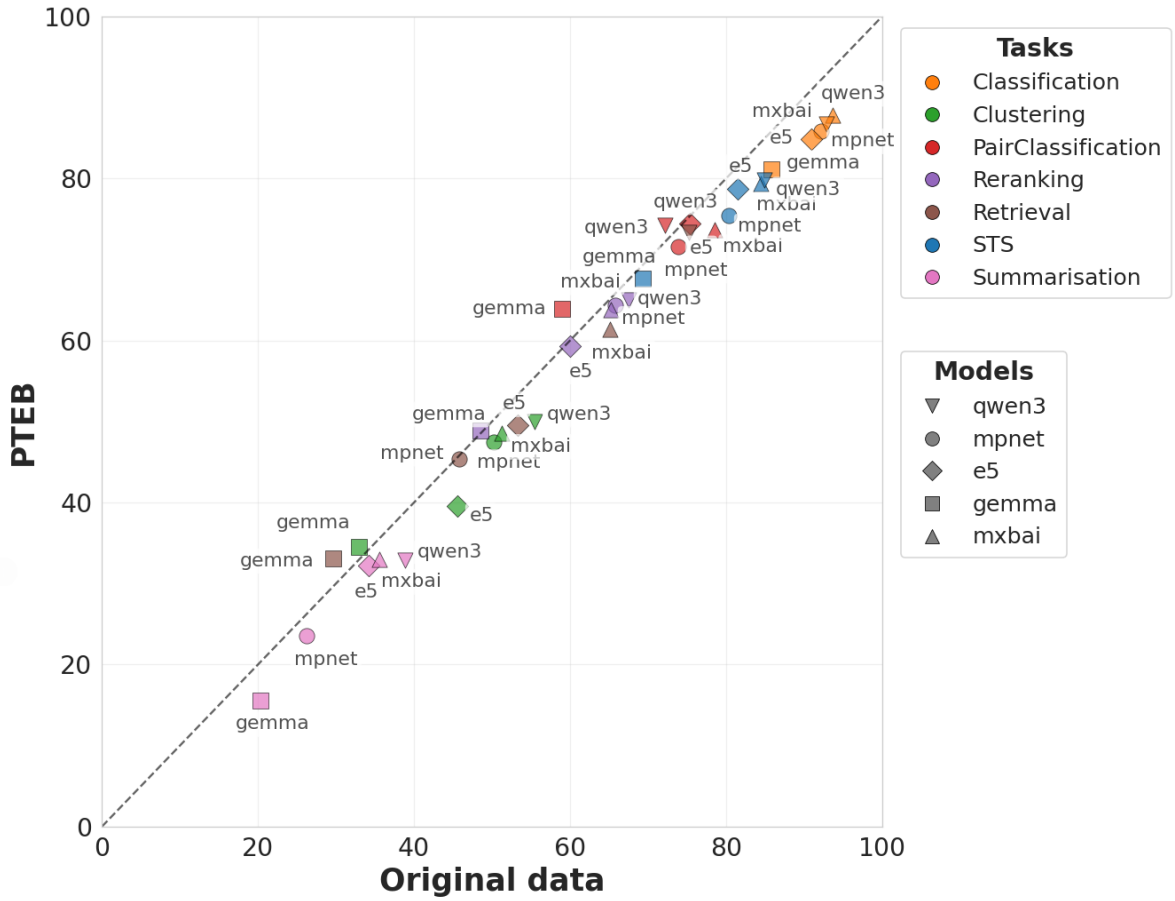

Figure 7: Scores per task on original MTEB data vs. PTEB (English only, for original MTEB data based on $n = 1$ runs, for PTEB based on $n = 6$ runs; in %).

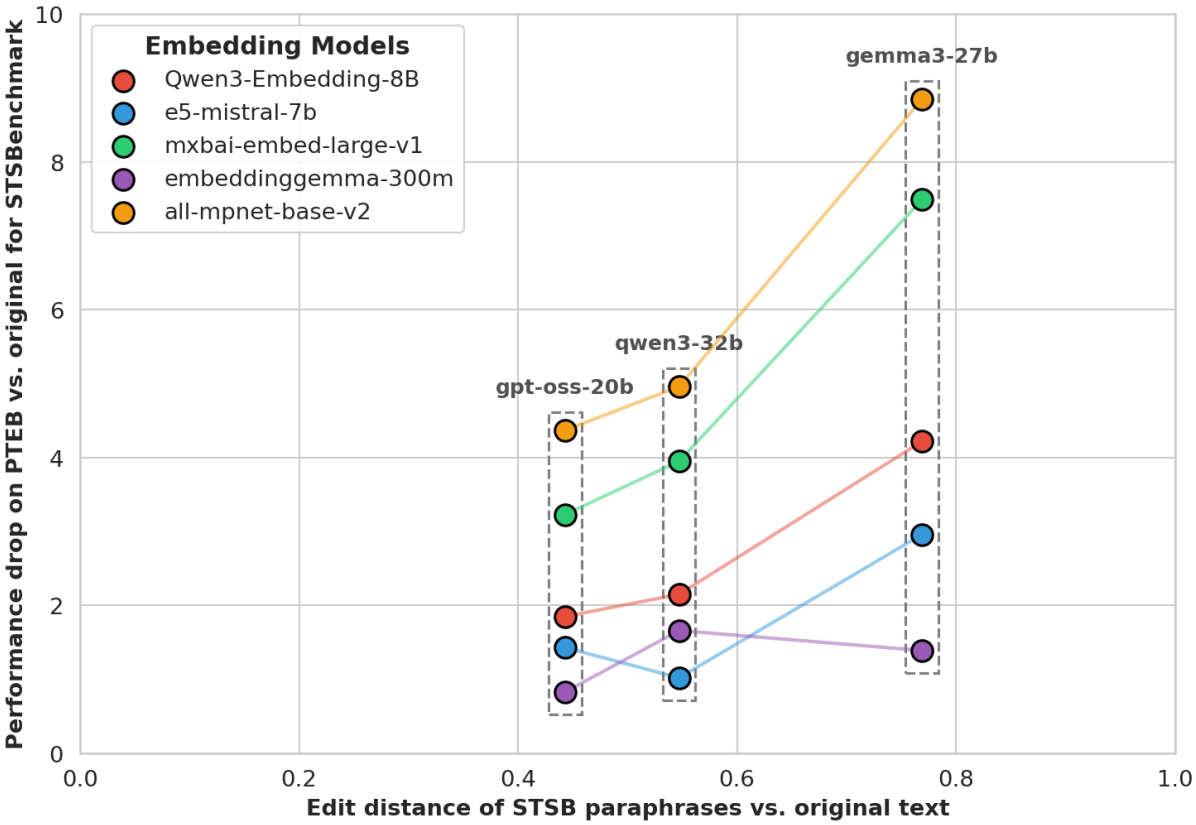

Figure 8: Edit distance of the paraphrases generated by the different generative paraphrase models (X-axis) vs. the performance drop of the embedding models on PTEB compared to original data for STSBenchmark ($n = 1$ runs; in %).