One prompt is not enough: Instruction Sensitivity Undermines Embedding Model Evaluation

Anonymous ACL submission

Abstract

Instruction embedding models have become common among state-of-the-art models, however are evaluated using a single prompt per task. The single-point evaluation ignores a main problem of the instruction-based approach namely: sensitivity to the phrasing of the instruction. We present an empirical study of prompt sensitivity across 6 embedding models, 11 datasets, and 15 task-specific prompts per dataset, a total of 990. We show that reported scores misrepresent the distribution of scores over plausible prompts. The default prompt can both systematically understate or overstate performance. Furthermore, we show that the leaderboard ranking is not robust to prompt selection: by choosing prompts favorably, any model in our study can be promoted to first place. Our findings suggest that single-prompt evaluation is insufficient for instruction-tuned embedding models and that benchmarks should incorporate prompt robustness, either by evaluating over multiple prompts or by reporting sensitivity alongside point estimates.

The code can be found in our GitHub repository[1]

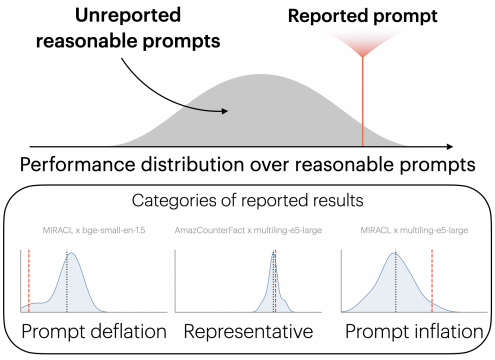

Figure 1: **Overview of our approach**, showing the currently unreported distribution of performances over reasonable prompts against the point estimate that is the reported results. Below that we show three possible option, either the reported result is representative of the distribution, it is lower than the (prompt deflation) or higher than (prompt inflation).

1 Introduction and Background

Embeddings are foundational to modern NLP applications, serving as the backbone for tasks such as Retrieval-Augmented Generation (RAG), semantic search, zero-shot clustering as well as newer tasks such as agentic memory (Lewis et al., 2020; Xu et al., 2026). Recently, instruction-tuned embedding models initially popularized by Su et al. (2023) have become common among top-performing embedding models such as Qwen or the e5-series (Zhang et al., 2025; Wang et al., 2024; Enevoldsen et al., 2025). Instructions (or prompts) enables models to capture task-specific nuances of the text, offering higher flexibility for various domains and objectives. However, while instructions allow for a high degree of user-flexibility they also allow instructions to be tuned to obtain better benchmark performance, thus posing a fundamental challenge of how to best measure, report and compare such models that do not oversell nor undersell their capabilities.

The most common benchmarks for evaluating the quality of embedding are implemented using the `beir` for information retrieval (Thakur et al., 2021; Kamalloo et al., 2024) and multitask `mteb` frameworks, which includes popular benchmarks such as MTEB, MMTEB (Muennighoff et al., 2022; Enevoldsen et al., 2025). `mteb` evaluate instruction-based models using only a single prompt either supplied by the task or the model contributor. This approach ignores a defining characteristic of these models: that they are sensitive to phrasing of the instruction. Relying on a single prompt risks misrepresenting actual model performance. Furthermore, it is possible to overwrite

[1] https://anonymous.4open.science/r/instruction-sensitivity-evaluation-F0F9/

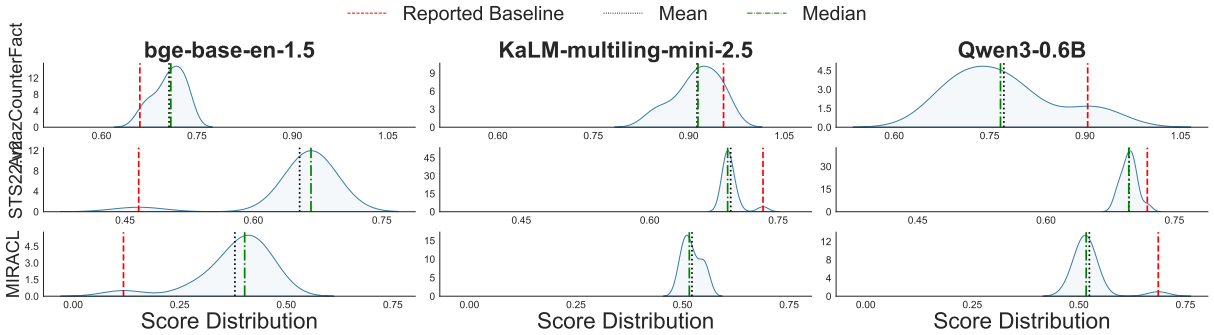

Figure 2: **Example of prompt deflation and inflation** on a sample of evaluated tasks and models. For all the scores and distributions see Appendix A.

the prompt within the models implementation – to allow customization to a specific prompt scheme – making it possible for the model developers to selectively choose the best performing prompts for a given task.

Current research on prompt sensitivity is focused mostly on generative LLMs (Wahle et al., 2024; Leidinger et al., 2023; Romanou et al., 2026; Mizrahi et al., 2024), while instruction-based embedding models remains largely underexplored. While recent efforts have addressed overfitting via paraphrases (Frank and Afli, 2026), no work systematically examines the effect of prompt variation on embedding evaluation. Our work bridges this gap by isolating the prompt as the primary variable to quantify its impact on reported performance.

In this paper, we question the validity of single-prompt evaluation for instruction-tuned embedding models. We investigate the following research questions:

> (**RQ1**) To what extent are instruction-tuned embedding models sensitive to variations of prompts relevant to the task?
> (**RQ2**) How do these variations influence the reported benchmark rankings?
> (**RQ3**) To what degree is benchmark hacking facilitated by prompt hacking (Qiu et al., 2026)?

To answer these questions, we conduct an empirical study across 11 diverse tasks types including retrieval, classification, clustering, and semantic textual similarity (STS). For each task, we synthesize a set of 15 task-specific prompts and evaluate a suite of different open-weight embedding models, which resulted in 165 prompts. Our findings reveal that reported benchmark scores frequently misrepresent expected performance: the default prompt can act as an outlier, artificially deflating or inflating a model's perceived quality (Figure 1). More critically, we show that by adversarially selecting prompts, any model in our study can be made to appear as the top-ranked model on a simulated leaderboard.

We recommend that benchmarks transition from single-point estimates to multi-prompt evaluation — reporting score distributions or robustness metrics alongside point estimates — and that model developers either optimize for prompt robustness or adopt limited prompt schemes such as those used by the Jina v5 series (Akram et al., 2026).

2 Methodology

To test the influence of the prompts on the benchmark performance, we selected the following diverse list of tasks and datasets from `mteb`[2]: retrieval (MIRACLRetrievalHardNegatives.v2, Touche2020Retrieval.v3, FEVERHardNegatives (Zhang et al., 2023; Thakur et al., 2024; Thorne et al., 2018)), classification (TweetSentimentClassification, ImdbClassification, AmazonCounterfactualClassification (Barbieri et al., 2022; Maas et al., 2011; O'Neill et al., 2021)), clustering (MedrxivClusteringP2P.v2, StackExchangeClustering.v2 (Enevoldsen et al., 2025; Geigle et al., 2021)), and semantic similarity (STS15, STS14, STS22.v2 (Biçici, 2015; Bandhakavi et al., 2014; Chen et al., 2022)). We chose tasks either derived from MTEB (Muennighoff et al., 2022) or MMTEB (Enevoldsen et al., 2025). To allow for comparison to the English only BGE models we only evaluate on the English subsets.

For each task, we generated a list of 15 prompts with gpt-oss-120b (OpenAI, 2025) with vLLM

[2]We use the dataset name in `mteb` to ensure reproducibility.

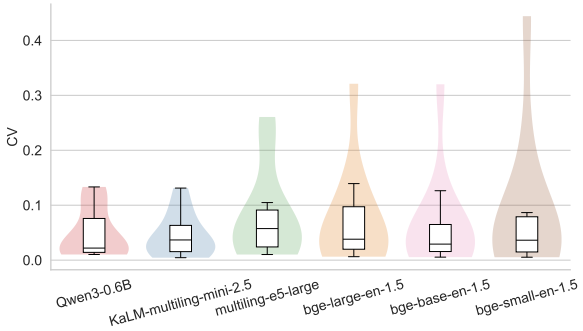

Figure 3: **Prompt sensitivity**: Coefficient of variance distributions across the models and prompts. We see that certain models have a high degree of variance across prompts.

(Kwon et al., 2023) structured outputs with the default parameters. The model was instructed to generate prompts based on the combination of language, task description, and task name. The prompts were manually evaluated to ensure that they were coherent and relevant for task (see Appendix C, and Appendix D for default prompts). The prompts generation was set up to resemble a real use-case, where the practitioner will rely on a custom, task-oriented prompt. We use synthetic prompt generation to reflect real practitioner behaviour and to avoid author bias in the reported effects. The resulting prompts are short (mean 5.24 words, std 2.33), consistent with documented model usage examples. We validated that synthetic and human-written prompts produce largely comparable score distributions, and that where differences exist they do not affect our conclusions (see Appendix C).

We seek to evaluate a broad selection including both multilingual models: Qwen3-Embedding-0.6B (Zhang et al., 2025), multilingual-e5-large-instruct (Wang et al., 2024), KaLM-embedding-multilingual-mini-instruct-v2.5 (Hu et al., 2025), as well as English models across a variety of sizes; bge-small-en-v1.5, bge-base-en-v1.5, and bge-large-en-v1.5 (Xiao et al., 2023). We ran the models on each task for each of the task-specific prompts, and compared the results with the reported scores on `mteb`. Models revisions and versions are available in Appendix B.

3 Analysis and Discussion

3.1 Prompt Inflation and Deflation

We examine whether the reported single-prompt results are representative of a model's expected performance. In the Appendix A (Figure 6), the score distributions across the prompts for each model/task combination are presented, with the reported `mteb` score.

We observe two systematic patterns. First, several models exhibit *prompt deflation*: the reported score is much lower that what would be expected. For example, BGE models show prompt deflation on TweetSentimentClassification, AmazonCounterfactualClassification, and STS22.v2 tasks (see Figure 2), where its reported scores are in the lower tail of the distributions. In a few cases we even see the reported MTEB score is drastically below the main distribution. This suggests that the default prompts used for these models on these tasks are suboptimal and does not correspond to the expected performance.

Second, we observe *prompt inflation*: the reported score falls in the upper tail of the distribution, overstating expected performance. For example, the reported score of Qwen3 on MIRACL (see Figure 2) task aligns with a secondary mode in the upper range of the distribution, which is above the primary peak. This indicates that the default prompt produces an atypically strong result.

3.2 Prompt Sensitivity Analysis

To answer RQ1, we analyze distributions of scores and coefficients of variation (CVs) of the scores obtained by the tasks per model (see Figure 3). The higher the CV is, the less robust the model is for variations in the prompt. We chose CV as it is unit-free, which allows us to compare scores for different tasks. While median CVs are moderate (2–6%), the distributions exhibit long tails: for the BGE models, individual task CVs reach 30–45%, confirming that specific model–task combinations are highly sensitive to prompt choice.

3.3 Effect on Leaderboard Rankings

Following RQ2, we investigate whether prompt sensitivity can be exploited to alter model rankings. Since `mteb` allows model contributors to specify custom prompts, a model developer could select a prompt that maximizes their model's score. To simulate this, we construct adversarial leaderboard configurations: for each target model, we assign it its best-performing prompt across tasks while assigning all other models either their worst-performing or default prompts, and compute the aggregated Borda rank. The rank changes per configuration are presented on Figure 4.

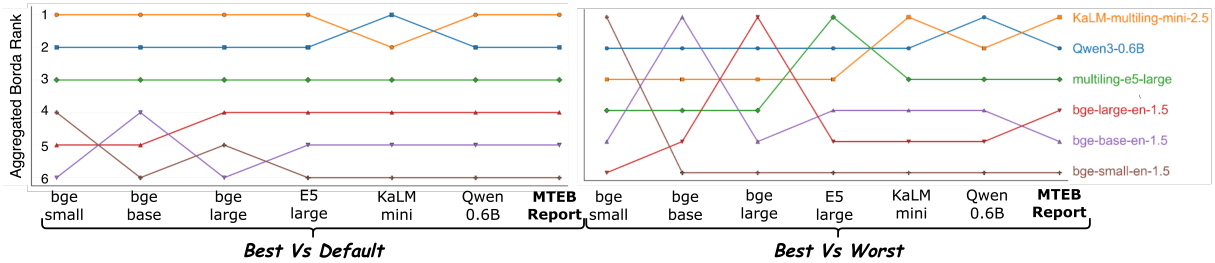

Figure 4: **Leaderboard sensitivity under adversarial prompt selection**. We show for each model two scenarios. Left: The model uses its best prompt while others use the default. Right: The model uses its best prompt while others use their worst. Ranks are computed via Borda scores across all tasks. In both cases we compare against the default configuration (MTEB Reported).

Under default prompts, KaLM and Qwen3 occupy ranks 1–2, e5-large rank 3, and the BGE variants ranks 4–6. However, under adversarial selection, *every model in our study can be promoted to rank 1* — including bge-small, which rises from last place when given its best prompts against competitors' worst. This scenario is not purely hypothetical: a company benchmarking models for deployment may naturally report their chosen model's best-case prompt while competitors' scores reflect arbitrary defaults, producing a misleading comparison.

Importantly, the milder scenario with the best performing prompt for the target model and default for others, still produces meaningful rank changes. For example, bge-large rises from rank 5 to rank 3 under this configuration, overtaking e5-large. This demonstrates that even a single model contributor optimizing their prompt, without any adversarial intent toward competitors, can distort the leaderboard.

3.4 Prompt Inflation in Reported Scores

Lastly, to examine whether some models systematically overfit (RQ3), we analyzed how the reported MTEB scores correspond to an observed score distribution. For each model–task combination, we calculated the empirical probability of the reported score across obtained scores (see Figure 5). Our results reveal a systematic issue: most models consistently report performance well above their expected scores, suggesting that favorable prompt selection is common in practice. This pattern is analogous to p-hacking (Simmons et al., 2011) in that it exploits undisclosed degrees of freedom in evaluation without constituting an explicit violation of submission policies — we term this *prompt hacking* (Qiu et al., 2026): the selective reporting of a single favorable prompt while remaining compliant, without modifying model weights, training data, or architecture. Importantly, prompt hacking need not be intentional: just as analytical bias can arise from well-meaning researchers exploring analytical choices without recognising the inflationary effect (Gelman and Loken, 2016), a model developer may simply report the prompt that worked best during development without deliberate intent to game the benchmark.

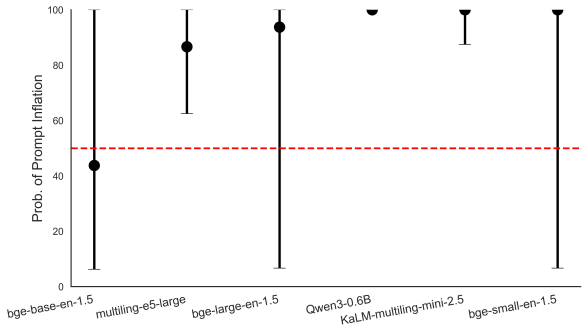

Figure 5: **Probability of prompt inflation**, i.e. reported score higher than the median, across tasks. The uncertainty is a 95% bootstrapped confidence interval.

4 Conclusion

In this paper, we conducted an empirical study of prompt sensitivity across 165 task-specific prompts. We find that reported benchmark scores frequently misrepresent expected performance: the MTEB default prompt can systematically overstate or understate a model's quality. Under adversarial prompt selection, every model in our study can be promoted to rank 1, including models that rank last under default evaluation. Even a single developer optimizing their prompt can shift rankings by multiple positions. We recommend that benchmarks adopt multi-prompt evaluation or similar strategies to ensure that the reported rankings faithfully reflect the models' quality.

Limitations

Model scope: We evaluate 6 open-weight models with up to 0.6B parameters. Larger instruction-tuned embedding models and closed-source models (e.g., OpenAI text-embedding-3, Cohere embed-v4) may exhibit different sensitivity profiles.

Prompt generation and sample size: Our prompt sets consist of 15 synthetically generated prompts per task, produced by a single language model. This is a relatively small sample: variance estimates (e.g., CV) computed over 15 points carry non-trivial uncertainty, and the synthetic generation process may not fully capture the diversity of prompts that practitioners write in practice. A larger, more diverse prompt set would yield more robust conclusions.

Language coverage: We restrict our analysis to English. Multilingual settings may introduce additional sources of prompt sensitivity related to language-specific phrasing conventions and translation artifacts.

Prompt variability as a practitioner asset: Our study frames prompt sensitivity primarily as a threat to benchmark validity, but the same variability can be an asset in practice. A developer who tries even a small set of task-specific prompts — as our results show — can gain up to 10 percentage points in performance at essentially no cost. This work does not evaluate strategies for exploiting this, such as prompt search or ensembling, and the optimal number of prompts to try in a practical setting remains an open question.

References

Mohammad Kalim Akram, Saba Sturua, Nastia Havriushenko, Quentin Herreros, Michael Günther, Maximilian Werk, and Han Xiao. 2026. jina-embeddings-v5-text: Task-targeted embedding distillation. *arXiv preprint arXiv:2602.15547*.

Anthropic. 2026. Claude. https://claude.ai/. [Large language model].

Anil Bandhakavi, Nirmalie Wiratunga, Deepak P, and Stewart Massie. 2014. Generating a word-emotion lexicon from #emotional tweets. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*, pages 12–21, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. XLM-T: Multilingual language models in Twitter for sentiment analysis and beyond. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 258–266, Marseille, France. European Language Resources Association.

Ergun Biçici. 2015. RTM-DCU: Predicting semantic similarity with referential translation machines. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 56–63, Denver, Colorado. 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.

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. *arXiv preprint arXiv:2502.13595*.

Manuel Frank and Haithem Afli. 2026. PTEB: Towards robust text embedding evaluation via stochastic paraphrasing at evaluation time with LLMs. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2832–2851, Rabat, Morocco. Association for Computational Linguistics.

Gregor Geigle, Nils Reimers, Andreas Rücklé, and Iryna Gurevych. 2021. Tweac: Transformer with extendable qa agent classifiers. *arXiv preprint*, abs/2104.07081.

Andrew Gelman and Eric Loken. 2016. The statistical crisis in science. *The best writing on mathematics (Pitici M, ed)*, 102:305–318.

Xinshuo Hu, Zifei Shan, Xinping Zhao, Zetian Sun, Zhenyu Liu, Dongfang Li, Shaolin Ye, Xinyuan Wei, Qian Chen, Baotian Hu, Haofen Wang, Jun Yu, and Min Zhang. 2025. Kalm-embedding: Superior training data brings a stronger embedding model. *Preprint*, arXiv:2501.01028.

Ehsan Kamalloo, Nandan Thakur, Carlos Lassance, Xueguang Ma, Jheng-Hong Yang, and Jimmy Lin. 2024. Resources for brewing beir: Reproducible reference models and statistical analyses. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 1431–1440, New York, NY, USA. Association for Computing Machinery.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E.

Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Alina Leidinger, Robert van Rooij, and Ekaterina Shutova. 2023. The language of prompting: What linguistic properties make a prompt successful? *Preprint*, arXiv:2311.01967.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

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.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv 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. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Xiaoyun Qiu, Yang Yu, and Haifeng Xu. 2026. On benchmark hacking in ml contests: Modeling, insights and design. *arXiv preprint arXiv:2604.22230*.

Angelika Romanou, Mark Ibrahim, Candace Ross, Chantal Shaib, Kerem Oktar, Samuel J. Bell, Anaelia Ovalle, Jesse Dodge, Antoine Bosselut, Koustuv Sinha, and Adina Williams. 2026. Brittlebench: Quantifying llm robustness via prompt sensitivity. *Preprint*, arXiv:2603.13285.

Joseph P. Simmons, Leif D. Nelson, and Uri Simonsohn. 2011. False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, 22(11):1359–1366.

Aaditya Singh, Adam Fry, and et al. 2025. Openai gpt-5 system card. *Preprint*, arXiv:2601.03267.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. *Preprint*, arXiv:2212.09741.

Nandan Thakur, Luiz Bonifacio, Maik Fröbe, Alexander Bondarenko, Ehsan Kamalloo, Martin Potthast, Matthias Hagen, and Jimmy Lin. 2024. Systematic evaluation of neural retrieval models on the Touché 2020 argument retrieval subset of BEIR. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Jan Philip Wahle, Terry Ruas, Yang Xu, and Bela Gipp. 2024. Paraphrase types elicit prompt engineering capabilities. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11004–11033, Miami, Florida, USA. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2026. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604.

Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2023. MIRACL: A Multilingual Retrieval Dataset Covering 18 Diverse Languages. *Transactions of the Association for Computational Linguistics*, 11:1114–1131.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

A Detailed Scores

B Model Revisions

For our experiments, we used the following model and revisions from HuggingFace Hub (see Table 1) (Wolf et al., 2020) and MTEB version 2.12.30.

C Prompt Evaluation

We manually evaluated the generated prompts. For each prompt, we looked at the task and task description, and assigned one of the following labels:

- Acceptable. The prompt sounds natural, relevant to the task, and the task is clearly stated.
- Irrelevant. The prompt is irrelevant to the task.
- Unnatural. The prompt does not sound correct and is not natural.

Based on our analysis *all* the generated prompts were considered to be acceptable. We noticed that some prompts contained name of the task in them, e.g. "Similarity score request, eng, SemEval STS 2014:". We marked these prompts as acceptable, as such prompts contain reasonable instructions, and the task name may provide an indication of the task to the model. In the case of the example, SemEval is a well known shared task, therefore the model may potentially know its relation towards STS.

To show that our findings are generalizable to natural human prompts, we created 10 prompts per task for BGE family of models and compared the human prompts performance with the machine generated prompts. The results are presented on Figure 7 and Figure 8. The distributions were tested with the permutation test on similarity of distributions.

The figure shows that the human and synthetic prompts perform similarly in most cases. In the cases, where the test did not reject the hypothesis of distributional similarity, the analysis of the reported score and performance distribution does not contradict findings of the paper.

D Default MTEB Prompts

In Table 2, Table 3, and Table 4 the default MTEB prompts are present per each model–task combination.

E Best and Worst Performing Prompts

F AI Tools Usage

In this work, we utilized AI tools to assist with code generation, debugging, and spell-checking/grammatical editing of the manuscript text. Specifically, we used Anthropic's Claude models (Anthropic, 2026) and OpenAI's ChatGPT (Singh et al., 2025).

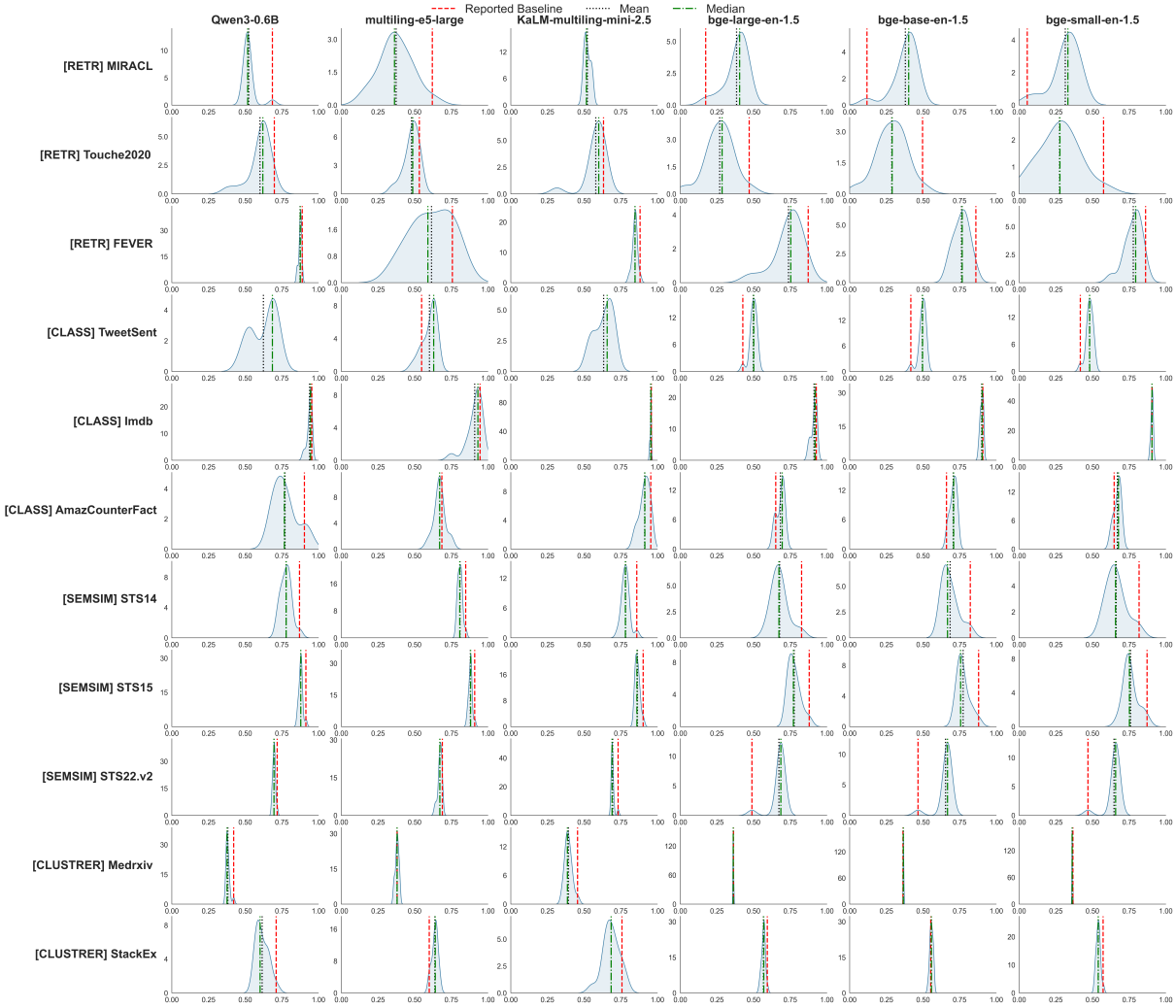

Figure 6: Detailed scores per model, metric, and task. For some tasks MTEB does not currently report a score hence the red line is missing.

Model	Commit
Qwen/Qwen3-Embedding-0.6B	97b0c614
intfloat/multilingual-e5-large-instruct	274baa43
KaLM-Embedding/KaLM-embedding-multilingual-mini-instruct-v2.5	52c687bb
BAAI/bge-small-en-v1.5	5c38ec7c
BAAI/bge-base-en-v1.5	a5beb1e3
BAAI/bge-large-en-v1.5	d4aa6901

Table 1: Model revisions used in experiments.

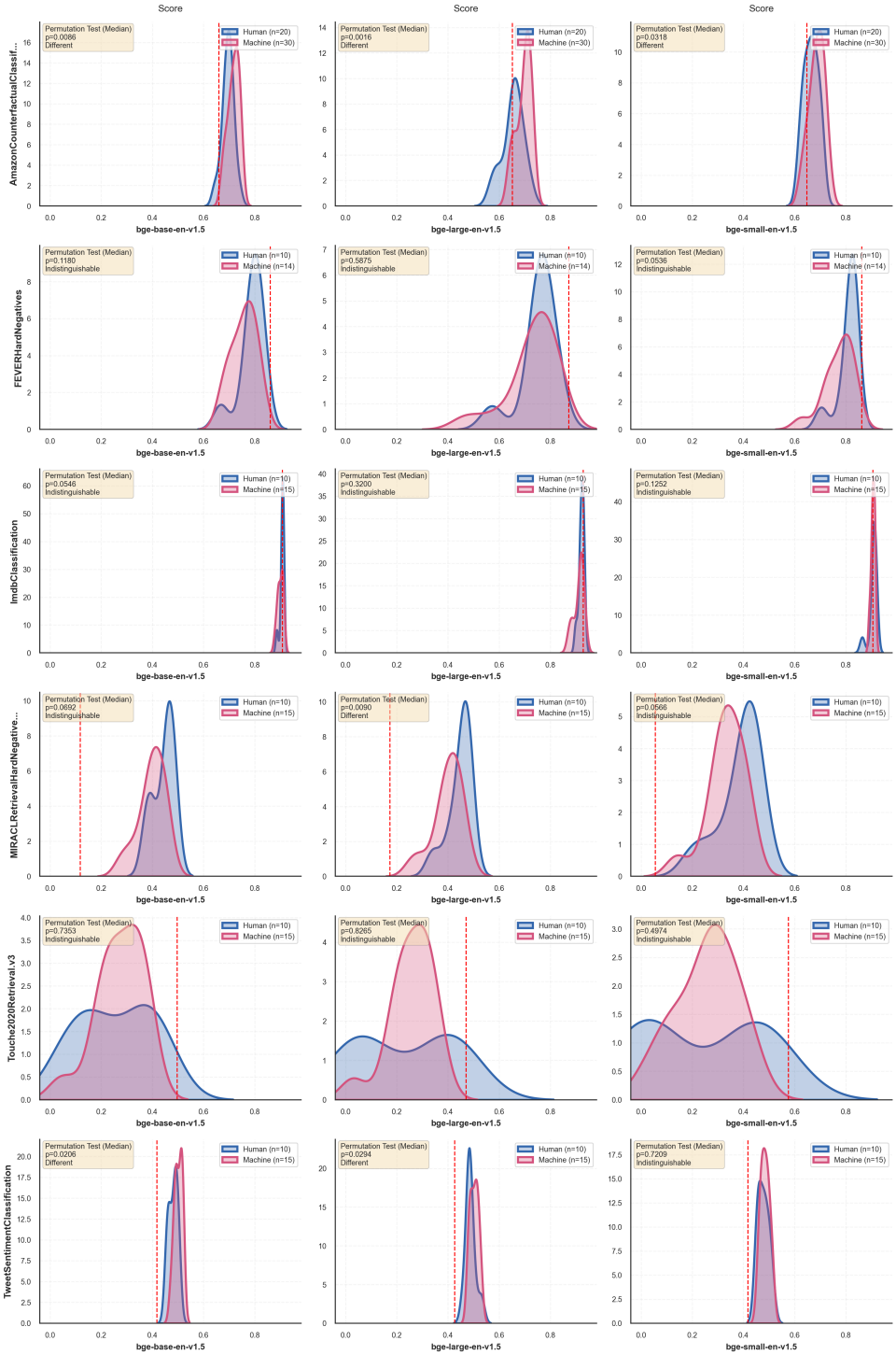

Figure 7: Comparison of human vs machine generated prompt score distribution for Classification and Retrieval tasks. Red dashed line indicates a reported score on MTEB.

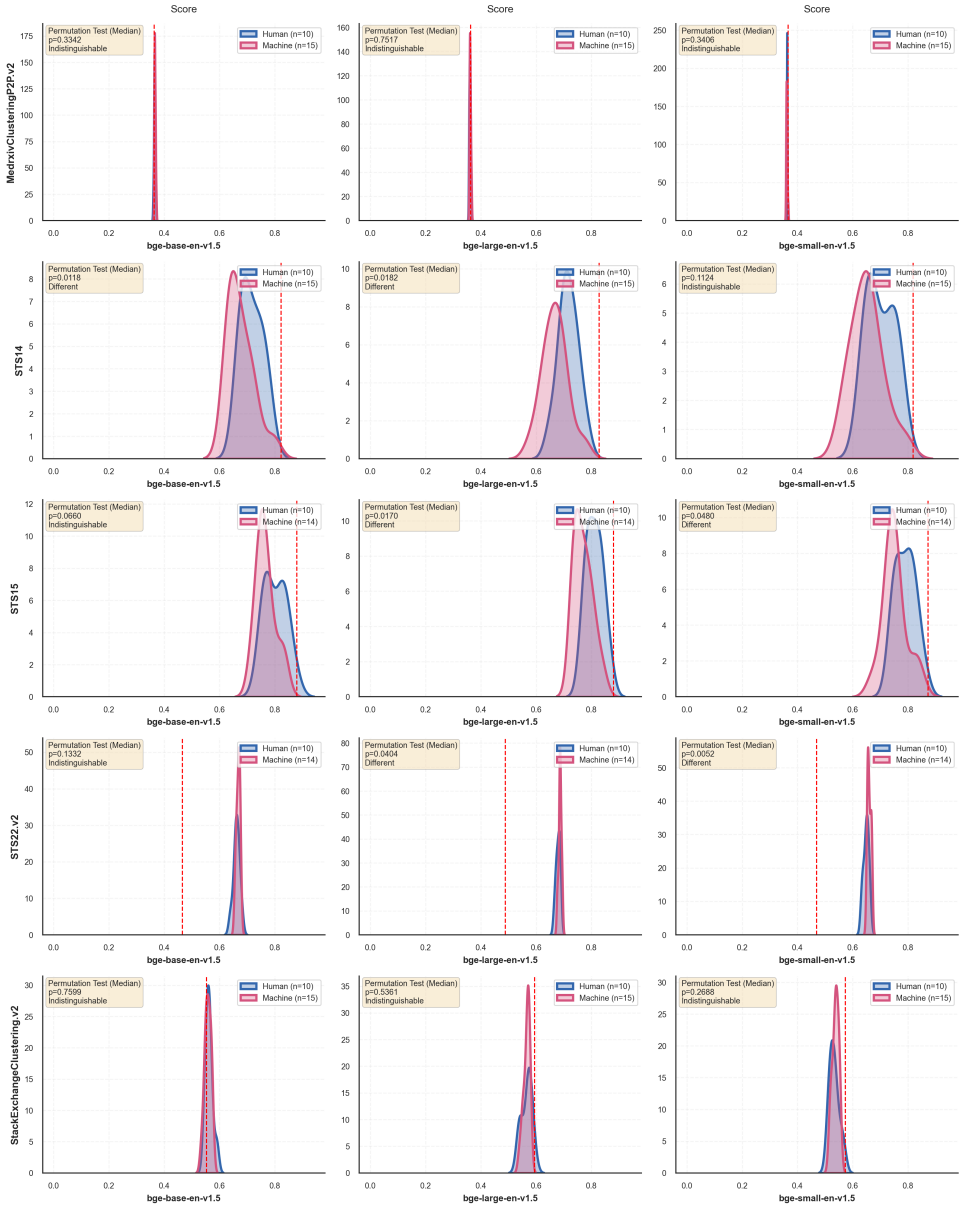

Figure 8: Comparison of human vs machine generated prompt score distribution for Clustering and STS tasks. Red dashed line indicates a reported score on MTEB.

Table 2: Generated prompts by model and task.

Model	Task	Prompt
Qwen/Qwen3-Embedding-0.6B	AmazonCounterfactualClass	Classify a given Amazon customer review text as either counterfactual or not-counterfactual
Qwen/Qwen3-Embedding-0.6B	FEVERHardNegatives	Retrieve text based on user query.
Qwen/Qwen3-Embedding-0.6B	STS22.v2	Retrieve semantically similar text.
Qwen/Qwen3-Embedding-0.6B	Touche2020Retrieval.v3	Retrieve text based on user query.
Qwen/Qwen3-Embedding-0.6B	STS15	Retrieve semantically similar text.
Qwen/Qwen3-Embedding-0.6B	STS14	Retrieve semantically similar text.
Qwen/Qwen3-Embedding-0.6B	MIRACLRetrievalHardNegatives	Given a question, retrieve Wikipedia passages that answer the question
Qwen/Qwen3-Embedding-0.6B	TweetSentimentClassification	Classify user passages.
Qwen/Qwen3-Embedding-0.6B	MedrxivClusteringP2P.v2	Identify the main category of Medrxiv papers based on the titles and abstracts
Qwen/Qwen3-Embedding-0.6B	StackExchangeClustering.v2	Identify the topic or theme of StackExchange posts based on the titles
Qwen/Qwen3-Embedding-0.6B	ImdbClassification	Classify the sentiment expressed in the given movie review text from the IMDB dataset
BAAI/bge-small-en-v1.5	AmazonCounterfactualClass	Represent this sentence for searching relevant passages:
BAAI/bge-small-en-v1.5	FEVERHardNegatives	Represent this sentence for searching relevant passages:
BAAI/bge-small-en-v1.5	STS22.v2	Represent this sentence for searching relevant passages:
BAAI/bge-small-en-v1.5	Touche2020Retrieval.v3	Represent this sentence for searching relevant passages:
BAAI/bge-small-en-v1.5	STS15	Represent this sentence for searching relevant passages:
BAAI/bge-small-en-v1.5	STS14	Represent this sentence for searching relevant passages:
BAAI/bge-small-en-v1.5	MIRACLRetrievalHardNegatives	Represent this sentence for searching relevant passages:
BAAI/bge-small-en-v1.5	TweetSentimentClassification	Represent this sentence for searching relevant passages:
BAAI/bge-small-en-v1.5	MedrxivClusteringP2P.v2	Represent this sentence for searching relevant passages:
BAAI/bge-small-en-v1.5	StackExchangeClustering.v2	Represent this sentence for searching relevant passages:
BAAI/bge-small-en-v1.5	ImdbClassification	Represent this sentence for searching relevant passages:
BAAI/bge-base-en-v1.5	AmazonCounterfactualClass	Represent this sentence for searching relevant passages:
BAAI/bge-base-en-v1.5	FEVERHardNegatives	Represent this sentence for searching relevant passages:
BAAI/bge-base-en-v1.5	STS22.v2	Represent this sentence for searching relevant passages:

Table 3: Generated prompts by model and task.

Model	Task	Prompt
BAAI/bge-base-en-v1.5	Touche2020Retrieval.v3	Represent this sentence for searching relevant passages:
BAAI/bge-base-en-v1.5	STS15	Represent this sentence for searching relevant passages:
BAAI/bge-base-en-v1.5	STS14	Represent this sentence for searching relevant passages:
BAAI/bge-base-en-v1.5	MIRACLRetrievalHardNegat	Represent this sentence for searching relevant passages:
BAAI/bge-base-en-v1.5	TweetSentimentClassificat	Represent this sentence for searching relevant passages:
BAAI/bge-base-en-v1.5	MedrxivClusteringP2P.v2	Represent this sentence for searching relevant passages:
BAAI/bge-base-en-v1.5	StackExchangeClustering.v	Represent this sentence for searching relevant passages:
BAAI/bge-base-en-v1.5	ImdbClassification	Represent this sentence for searching relevant passages:
BAAI/bge-large-en-v1.5	AmazonCounterfactualClass	Represent this sentence for searching relevant passages:
BAAI/bge-large-en-v1.5	FEVERHardNegatives	Represent this sentence for searching relevant passages:
BAAI/bge-large-en-v1.5	STS22.v2	Represent this sentence for searching relevant passages:
BAAI/bge-large-en-v1.5	Touche2020Retrieval.v3	Represent this sentence for searching relevant passages:
BAAI/bge-large-en-v1.5	STS15	Represent this sentence for searching relevant passages:
BAAI/bge-large-en-v1.5	STS14	Represent this sentence for searching relevant passages:
BAAI/bge-large-en-v1.5	MIRACLRetrievalHardNegat	Represent this sentence for searching relevant passages:
BAAI/bge-large-en-v1.5	TweetSentimentClassificat	Represent this sentence for searching relevant passages:
BAAI/bge-large-en-v1.5	MedrxivClusteringP2P.v2	Represent this sentence for searching relevant passages:
BAAI/bge-large-en-v1.5	StackExchangeClustering.v	Represent this sentence for searching relevant passages:
BAAI/bge-large-en-v1.5	ImdbClassification	Represent this sentence for searching relevant passages:
intfloat/multilingual-e5-	AmazonCounterfactualClass	Classify a given Amazon customer review text as either counterfactual or not-counterfactual
intfloat/multilingual-e5-	FEVERHardNegatives	Retrieve text based on user query.
intfloat/multilingual-e5-	STS22.v2	Retrieve semantically similar text.
intfloat/multilingual-e5-	Touche2020Retrieval.v3	Retrieve text based on user query.
intfloat/multilingual-e5-	STS15	Retrieve semantically similar text.
intfloat/multilingual-e5-	STS14	Retrieve semantically similar text.
intfloat/multilingual-e5-	MIRACLRetrievalHardNegat	Given a question, retrieve Wikipedia passages that answer the question
intfloat/multilingual-e5-	TweetSentimentClassificat	Classify user passages.
intfloat/multilingual-e5-	MedrxivClusteringP2P.v2	Identify the main category of Medrxiv papers based on the titles and abstracts
intfloat/multilingual-e5-	StackExchangeClustering.v	Identify the topic or theme of StackExchange posts based on the titles
intfloat/multilingual-e5-	ImdbClassification	Classify the sentiment expressed in the given movie review text from the IMDB dataset
KaLM-Embedding/KaLM-embed	AmazonCounterfactualClass	Classify a given Amazon customer review text as either counterfactual or not-counterfactual

Table 4: Generated prompts by model and task.

Model	**Task**	**Prompt**
KaLM-Embedding/KaLM-embed	FEVERHardNegatives	Retrieve text based on user query.
KaLM-Embedding/KaLM-embed	STS22.v2	Retrieve semantically similar text.
KaLM-Embedding/KaLM-embed	Touche2020Retrieval.v3	Retrieve text based on user query.
KaLM-Embedding/KaLM-embed	STS15	Retrieve semantically similar text
KaLM-Embedding/KaLM-embed	STS14	Retrieve semantically similar text
KaLM-Embedding/KaLM-embed	MIRACLRetrievalHardNegati	Given a question, retrieve Wikipedia passages that answer the question
KaLM-Embedding/KaLM-embed	TweetSentimentClassificat	Classify user passages.
KaLM-Embedding/KaLM-embed	MedrxivClusteringP2P.v2	Identify the main category of Medrxiv papers based on the titles and abstracts
KaLM-Embedding/KaLM-embed	StackExchangeClustering.v	Identify the topic or theme of StackExchange posts based on the titles
KaLM-Embedding/KaLM-embed	ImdbClassification	Classify the sentiment expressed in the given movie review text from the IMDB dataset

Table 5: Best and worst prompts for task: AmazonCounterfactualClassification

Model	Type	Score	Prompt
`BAAI/bge-base-en-v1.5`	Best	0.734	`Label this example:`
	Worst	0.66	`default`
`BAAI/bge-large-en-v1.5`	Best	0.719	`Label this example:`
	Worst	0.637	`Text classification (eng) - Amazon reviews:`
`BAAI/bge-small-en-v1.5`	Best	0.705	`Label this example:`
	Worst	0.62	`Assign a category to this input:`
`KaLM-Embedding/KaLM-embed...`	Best	0.955	`default`
	Worst	0.844	`Assign a category to this input:`
`Qwen/Qwen3-Embedding-0.6B`	Best	0.928	`Assign a counterfactual detection label:`
	Worst	0.674	`Text classification (eng) - Amazon reviews:`
`intfloat/multilingual-e5-...`	Best	0.75	`Assign a counterfactual detection label:`
	Worst	0.589	`Predict the class:`

Table 6: Best and worst prompts for task: FEVERHardNegatives

Model	Type	Score	Prompt
BAAI/bge-base-en-v1.5	Best	0.86	default
	Worst	0.668	Text to be used in retrieval:
BAAI/bge-large-en-v1.5	Best	0.873	default
	Worst	0.472	Search representation for retrieval:
BAAI/bge-small-en-v1.5	Best	0.862	default
	Worst	0.627	Search representation for retrieval:
KaLM-Embedding/KaLM-embed...	Best	0.882	default
	Worst	0.807	Encode this FEVER claim in eng for retrieval:
Qwen/Qwen3-Embedding-0.6B	Best	0.889	default
	Worst	0.853	Encode this claim for retrieval:
intfloat/multilingual-e5-...	Best	0.805	Retrieve relevant documents for the given claim:
	Worst	0.366	Encode this input for retrieval:

Table 7: Best and worst prompts for task: ImdbClassification

Model	Type	Score	Prompt
BAAI/bge-base-en-v1.5	Best	0.914	Assign a label to this review:
	Worst	0.878	Encode and classify the following eng snippet from Large Movie Review Dataset:
BAAI/bge-large-en-v1.5	Best	0.935	Assign a label to this review:
	Worst	0.874	Category prediction for the following eng review (Large Movie Review Dataset):
BAAI/bge-small-en-v1.5	Best	0.917	Assign a label to this review:
	Worst	0.89	Determine the class for the given passage:
KaLM-Embedding/KaLM-embed...	Best	0.959	default
	Worst	0.945	Determine the class for the given passage:
Qwen/Qwen3-Embedding-0.6B	Best	0.954	default
	Worst	0.894	Determine the class for the given passage:
intfloat/multilingual-e5-...	Best	0.95	Classify the movie review:
	Worst	0.75	Determine the class for the given passage:

Table 8: Best and worst prompts for task: MIRACLRetrievalHardNegatives.v2

Model	Type	Score	Prompt
BAAI/bge-base-en-v1.5	Best	0.464	Search for documents using this query:
	Worst	0.118	default
BAAI/bge-large-en-v1.5	Best	0.472	Retrieve relevant passage:
	Worst	0.173	default
BAAI/bge-small-en-v1.5	Best	0.421	Search passage for MIRACL in eng:
	Worst	0.055	default
KaLM-Embedding/KaLM-embed...	Best	0.556	Search for documents using this query:
	Worst	0.492	Retrieve relevant text in eng:
Qwen/Qwen3-Embedding-0.6B	Best	0.686	default
	Worst	0.491	Encode for retrieval in eng:
intfloat/multilingual-e5-...	Best	0.621	default
	Worst	0.163	Encode for retrieval in eng:

Table 9: Best and worst prompts for task: MedrxivClusteringP2P.v2

Model	Type	Score	Prompt
BAAI/bge-base-en-v1.5	Best	0.371	Cluster representation:
	Worst	0.363	default
BAAI/bge-large-en-v1.5	Best	0.367	Clustering input:
	Worst	0.358	Cluster these medrxiv titles+abstracts (eng):
BAAI/bge-small-en-v1.5	Best	0.367	default
	Worst	0.359	Group similar medrxiv texts into one of 51 categories:
KaLM-Embedding/KaLM-embed...	Best	0.456	default
	Worst	0.354	Group similar texts:
Qwen/Qwen3-Embedding-0.6B	Best	0.422	default
	Worst	0.37	Assign to a cluster (eng):
intfloat/multilingual-e5-...	Best	0.396	Cluster the medrxiv title and abstract:
	Worst	0.36	Group similar texts (eng):

Table 10: Best and worst prompts for task: STS14

Model	Type	Score	Prompt
BAAI/bge-base-en-v1.5	Best	0.823	default
	Worst	0.626	Encode for semantic matching:
BAAI/bge-large-en-v1.5	Best	0.828	default
	Worst	0.582	Text pair similarity assessment:
BAAI/bge-small-en-v1.5	Best	0.818	default
	Worst	0.562	Text pair similarity assessment:
KaLM-Embedding/KaLM-embed...	Best	0.86	default
	Worst	0.731	Assess semantic similarity, language eng, dataset SemEval STS 2014:
Qwen/Qwen3-Embedding-0.6B	Best	0.871	default
	Worst	0.723	Text pair similarity assessment:
intfloat/multilingual-e5-...	Best	0.848	default
	Worst	0.787	Encode this pair for similarity measurement:

Table 11: Best and worst prompts for task: STS15

Model	Type	Score	Prompt
BAAI/bge-base-en-v1.5	Best	0.88	default
	Worst	0.717	Encode pair for semantic-similarity in eng:
BAAI/bge-large-en-v1.5	Best	0.88	default
	Worst	0.733	Encode for semantic matching:
BAAI/bge-small-en-v1.5	Best	0.873	default
	Worst	0.674	Measure text similarity in eng, SemEval STS 2015:
KaLM-Embedding/KaLM-embed...	Best	0.903	default
	Worst	0.843	Measure text similarity in eng, SemEval STS 2015:
Qwen/Qwen3-Embedding-0.6B	Best	0.914	default
	Worst	0.857	Measure text similarity in eng, SemEval STS 2015:
intfloat/multilingual-e5-...	Best	0.91	default
	Worst	0.858	Text pair similarity assessment:

Table 12: Best and worst prompts for task: STS22.v2

Model	Type	Score	Prompt
BAAI/bge-base-en-v1.5	Best	0.678	In eng, assess similarity of news article pair - SemEval 2022 Task 8:
	Worst	0.466	default
BAAI/bge-large-en-v1.5	Best	0.694	Semantic similarity (eng) - SemEval 2022 Task 8 News:
	Worst	0.489	default
BAAI/bge-small-en-v1.5	Best	0.669	Evaluate semantic similarity (eng) on filtered SemEval 2022 Task 8 data:
	Worst	0.469	default
KaLM-Embedding/KaLM-embed...	Best	0.732	default
	Worst	0.688	Encode for semantic matching:
Qwen/Qwen3-Embedding-0.6B	Best	0.718	default
	Worst	0.682	Measure semantic similarity:
intfloat/multilingual-e5-...	Best	0.69	default
	Worst	0.638	Similarity score for article pair:

Table 13: Best and worst prompts for task: StackExchangeClustering.v2

Model	Type	Score	Prompt
BAAI/bge-base-en-v1.5	Best	0.573	Cluster representation:
	Worst	0.535	Cluster these stackexchange titles (eng):
BAAI/bge-large-en-v1.5	Best	0.594	default
	Worst	0.54	Cluster these stackexchange titles (eng):
BAAI/bge-small-en-v1.5	Best	0.573	default
	Worst	0.517	Encode for clustering, language eng:
KaLM-Embedding/KaLM-embed...	Best	0.778	Cluster these StackExchange titles:
	Worst	0.556	Group similar texts:
Qwen/Qwen3-Embedding-0.6B	Best	0.712	default
	Worst	0.565	Cluster representation:
intfloat/multilingual-e5-...	Best	0.672	Cluster these StackExchange titles:
	Worst	0.6	default

Table 14: Best and worst prompts for task: Touche2020Retrieval.v3

Model	Type	Score	Prompt
BAAI/bge-base-en-v1.5	Best	0.497	default
	Worst	0.048	Encode (eng) for Touché Task 1: Argument Retrieval for Controversial Questions:
BAAI/bge-large-en-v1.5	Best	0.471	default
	Worst	0.03	Encode (eng) for Touché Task 1: Argument Retrieval for Controversial Questions:
BAAI/bge-small-en-v1.5	Best	0.576	default
	Worst	0.021	Encode (eng) for Touché Task 1: Argument Retrieval for Controversial Questions:
KaLM-Embedding/KaLM-embed...	Best	0.637	Query for retrieval:
	Worst	0.315	Encode (eng) for Touché Task 1: Argument Retrieval for Controversial Questions:
Qwen/Qwen3-Embedding-0.6B	Best	0.699	default
	Worst	0.383	Encode (eng) for Touché Task 1: Argument Retrieval for Controversial Questions:
intfloat/multilingual-e5-...	Best	0.543	Retrieve arguments for controversial question:
	Worst	0.349	Encode (eng) for Touché Task 1: Argument Retrieval for Controversial Questions:

Table 15: Best and worst prompts for task: TweetSentimentClassification

Model	Type	Score	Prompt
BAAI/bge-base-en-v1.5	Best	0.518	Classify the following text:
	Worst	0.417	default
BAAI/bge-large-en-v1.5	Best	0.529	Label the tweet's sentiment:
	Worst	0.427	default
BAAI/bge-small-en-v1.5	Best	0.514	Label this example:
	Worst	0.417	default
KaLM-Embedding/KaLM-embed...	Best	0.704	Classify the sentiment of this tweet:
	Worst	0.534	Determine the class for the given passage:
Qwen/Qwen3-Embedding-0.6B	Best	0.706	Classify the sentiment of this tweet:
	Worst	0.489	Determine the class for the given passage:
intfloat/multilingual-e5-...	Best	0.645	eng sentiment analysis - classify:
	Worst	0.485	Determine the class for the given passage: