To MRL or Not To MRL: Comparing Random Vector Truncation Against Matryoshka Embeddings as Cost Reduction Methods for Text Encoders

Anonymous ACL submission

Abstract

Matryoshka Representation Learning (MRL) is a widely adopted approach for training text encoders so they provide useful text representations at various sizes, available by simply truncating the resulting vectors at sizes predetermined at training time. Recent works have shown that randomly truncating text embeddings has minimal impact in downstream performance unless vectors are reduced in size by at least 70%. However, random truncation has not yet been compared to MRL, so that it is unclear to what extent it is useful at reducing costs in applications that rely on text encoders. In this short paper, we benchmark random truncation applied to models that were trained with and without MRL. Our results across several models and downstream tasks show that, unless heavily truncating embeddings (i.e. reducing their size by at least 80%), randomly truncated embeddings of non-MRL models are at least competitive, and often outperform models trained with MRL. This suggests that random truncation is indeed a highly effective method of embedding reduction, even when compared to MRL, and that it is unclear how to best train models with MLR, as the additional training costs only become beneficial at very high truncation levels. Our code is attached to our ARR submission.

1 Introduction

Text embeddings have been widely adopted in many NLP tasks, from retrieval (Huang et al., 2020) to recommender systems (Zhao et al., 2023) to many others (Zhao et al., 2024). To reduce costs while retaining performance in such tasks, the use of small but well-performing embeddings is desirable. E.g. in text retrieval, bi-encoder models with less than 1B parameters are a common choice for first-stage retrieval over an entire collection of documents (Zhao et al., 2024). To provide more flexibility in this regard, Matryoshka Representation Learning (MRL) (Kusupati et al., 2022) is an approach that adds additional terms to the training objective so that text encoders are able to provide useful representations at various embedding sizes, available by simply truncating the resulting vectors at sizes pre-determined during training. Training models with MRL is already widely adopted, with some of the latest models providing this benefit out-of-the-box, such as Jina-Embeddings-V3 (Sturua et al., 2024), Qwen3-Embedding (Zhang et al., 2025b), and EmbeddingGemma (Vera et al., 2025). Recent works have empirically shown that randomly truncating text embeddings is a promising approach to obtaining vectors of reduced size even without specifically training encoders for it, as the impact in performance was minimal unless vector sizes were reduced by at least 70% (Tsukagoshi and Sasano, 2025; Takeshita et al., 2025; Inkiriwang et al., 2025). However, none of these studies compared random truncation to MRL, so that it is unclear to what extent random truncation is a useful approach for trading off performance for runtime and memory costs. In this short paper,

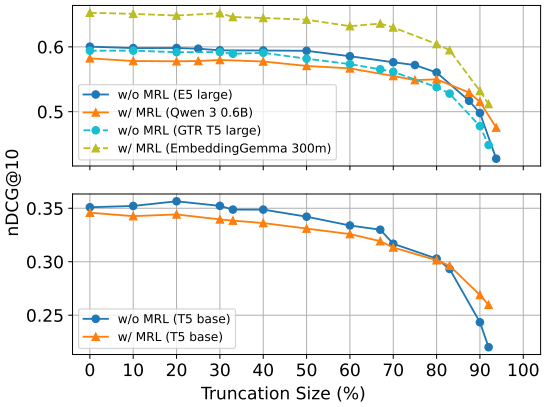

Figure 1: (Top) Performance of open text encoders as truncation levels increase looks the same whether trained with or without MRL. (Bottom) When models differ only in their use of MRL, truncation on non-MRL models is superior unless heavy truncation is applied.

we compare performance at various vector truncation levels using models trained with and without MRL. Our results show that unless heavily truncating embeddings, i.e. reducing their size by about 80%, randomly truncated embeddings of non-MRL models are at least competitive with, and often superior in performance compared to models trained with MRL. We found this across 4 open models, 10 newly trained text encoders and 24 embedding-based downstream tasks (e.g. see Fig. 1). These results suggest that random truncation is indeed a highly effective method for reducing many of the costs associated with text encoders, and that the additional costs of training with MRL are only beneficial in scenarios where reducing embedding size by about 80% or more is desirable. Further evidence of this is the fact that while MRL requires that truncation dimensions be chosen before training, MRL behaves the same as random truncations in that there is no difference in truncation performance on MRL models when truncating outside of those selected dimensions, suggesting further studies into the optimal way to train with MRL.

2 Background and Related Work

Matryoshka Representation Learning (MRL). Following Kusupati et al. (2022), we formalize MRL in a supervised setting. Let $f : T \rightarrow \mathbb{R}^d$ be a text encoder with parameters $\boldsymbol{\theta}$ that maps a sequence of tokens $x \in \mathcal{T}$ to a d-dimensional vector $\boldsymbol{e}$, i.e. $\boldsymbol{e} = f(x|\boldsymbol{\theta})$, where $\mathcal{T}$ is the set of all possible token sequences in a given language. Given labelled dataset $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ and loss function $\mathcal{L}$, MRL optimizes the following objective:

$$\min_{\boldsymbol{\theta}, \mathcal{W}_\mathcal{M}} \frac{1}{N} \sum_{(x,y)\in\mathcal{D}} \sum_{m\in\mathcal{M}} c_m \cdot \mathcal{L}(\boldsymbol{W}_m \cdot f_m(x|\boldsymbol{\theta}), y)$$

where $\mathcal{M} = \{m_1, m_2, \ldots, m_{|M|}\}$ is a set of vector sizes $m_i \leq d$, $\mathcal{W}_\mathcal{M} = \{\boldsymbol{W}_m\}_{m\in\mathcal{M}}$ the set of learnable classifiers for each $m \in \mathcal{M}$, c_m a scaling factor for the loss term corresponding to vector size m, and $f_m(x|\boldsymbol{\theta}) = f(x|\boldsymbol{\theta})_{1:m}$ the output vector of encoder f truncated to size m. In other words, MRL applies empirical risk minimization over the set of classifiers $\mathcal{W}_\mathcal{M}$, where each uses a different truncation of vector $\boldsymbol{e}$ as input. Weights are typically shared across all classifiers in $\mathcal{W}_\mathcal{M}$ for efficiency, i.e. a nested classifier is trained. Optionally, $\mathcal{W}_\mathcal{M}$ can be dropped so that only $\boldsymbol{\theta}$ is optimized.

Subsequent works have proposed variants of MRL, e.g. as a method to inject MRL properties to already trained models (Yoon et al., 2024), or as a form of reduction of the number of layers in a model (Wang et al., 2025). More recently, Zhang et al. (2025a) studied MRL in more detail and compared it to other methods for dimensionality reduction. However, to our knowledge, no such studies have used random truncation as a baseline.

Random truncation. Inspired by the fact that the proof in the seminal Johnson-Lindenstrauss lemma (Johnson et al., 1984) is based on random projections, many works have since based their embedding-based methods on random projections (Achlioptas, 2003). More recently, Takeshita et al. (2025) found that randomly truncating text embeddings results in minimal loss in performance unless vectors are heavily reduced in size. They found that this observation may not be explained by some geometric properties of the embedding space, such as anisotropy or outlier dimensions. In concurrent work, Tsukagoshi and Sasano (2025) also reported the same efficacy of using random truncation as a means of dimensionality reduction.

The closest work to ours is that of Inkiriwang et al. (2025). They compare the efficacy of several methods for dimensionality reduction, including random projections, a method which they report as one of the most consistently effective ones. However, while they include more dimensionality reduction methods, we include more recent models and a larger variety of downstream tasks. More importantly, they do not include MRL in their comparison, despite it being used by some of the best performing text encoders to date (Sturua et al., 2024; Zhang et al., 2025b; Vera et al., 2025).

3 MRL vs Random Truncation

3.1 Benchmarking Open Encoders

In this section, we compare publicly available text encoders trained with and without MRL. We note that while there are many more differences between these models (e.g. base architecture or number of parameters), our goal is to determine whether MRL makes a difference w.r.t. retaining more of the performance of their full-size embeddings after truncation compared to non-MRL models. For consistency with MRL, we perform random truncations by removing the last elements of the embeddings, though random truncation can be successfully applied in arbitrary ways (Takeshita et al., 2025).

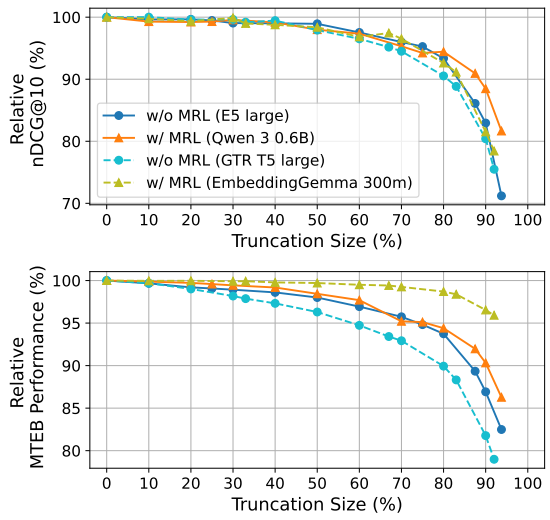

Figure 2: Performance on NanoBEIR (top) and MTEB (bottom) of text embeddings truncated at various sizes, relative to the performance of their corresponding full-sized embeddings, using four publicly available text encoders with and without MRL.

Models. We consider four seminal and state-of-the-art open-weight text encoders. For non-MRL models, we evaluate E5-Large (Wang et al., 2024) and GTR-T5-Large (Ni et al., 2022b). Both have similar number of parameters (600M), but E5-Large is based on an encoder-only model (BERT Large) while GTR-T5 is based on an encoder-decoder model (T5). For MRL models, we use Qwen3-Embedding-0.6B (Zhang et al., 2025b), a decoder-only-based model with 0.6B parameters, and EmbeddingGemma-300M (Vera et al., 2025), a encoder-decoder-based model.

Evaluation. We use 13 datasets from NanoBEIR[1], a smaller variant of BEIR (Thakur et al., 2021)), for retrieval evaluation. For classification, we use 11 datasets from MTEB (Muennighoff et al., 2023). See Table 1 in Appendix A.1 for a list of our datasets. As the original performances of these four models can vastly differ, we report the truncated embeddings' performance relative to the original full-sized embeddings.

Results and discussion. Our results are shown in Fig. 2. If MRL were to bring an advantage, we should see that performance of MRL models drops more slowly than non-MRL models. However, this is only true for EmbneddingGemma on MTEB, which suggests that MRL is indeed very effective for EmbeddingGemma, especially in contrast to GTR-T5, which drops much faster than all other models. However, this is not generally true, as EmbeddingGemma's advantage disappears in BEIR, where all models exhibit the same behavior, suggesting no advantage of MRL models over non-MRL ones, up until a truncation level between 60 and 70% (depending on dataset), where MRL models do become increasingly superior. In general, these results suggest that MRL is only beneficial when heavy truncation is applied. We also note MRL models show the same behavior as non-MRL models in that the pre-selected dimensions in MRL do not seem to be more beneficial than others.

3.2 Benchmarking Trained Encoders

While the experiments in the previous section allow us to study MRL using state-of-the-art text encoders, the various differences between models, and not just MRL, may account for the observed results. In this section, and to better determine the impact of MRL when using truncated embeddings, we train pairs of encoders where the only difference is the application of MRL. Unless otherwise noted, experimental settings are the same as in the previous section.

Models. We take five pre-trained language models (PLMs) from different sizes and architectures as starting checkpoints for our contrastive training. Each PLM will result in two text encoders, one with MRL and one without. We train BERT in two sizes (base and large) (Devlin et al., 2019) as well as RoBERTa (base and large) (Liu et al., 2020). Since these two are encoder-only PLMs, we complement the selection by adding T5 base Encoder (Raffel et al., 2020) to our training, as it originates from an encoder-decoder architecture.

Contrastive training. We train all PLMs with multiple negative ranking loss (Henderson et al., 2017) both for non-MRL and MRL models, a contrastive loss commonly used to train text encoders (Gao et al., 2021; Ni et al., 2022a; Vulić et al., 2023). For MRL models, we uniformly weigh the losses for each of the following nested dimensions: 64, 128, 256, 512 and 768. For training and validation data, we use the concatenation of SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018) as done in previous works (Reimers and Gurevych, 2019; Gao et al., 2021; Takeshita et al., 2025). We train models until convergence and select the best models on held-out

[1] https://huggingface.co/collections/zeta-alpha-ai/nanobeir

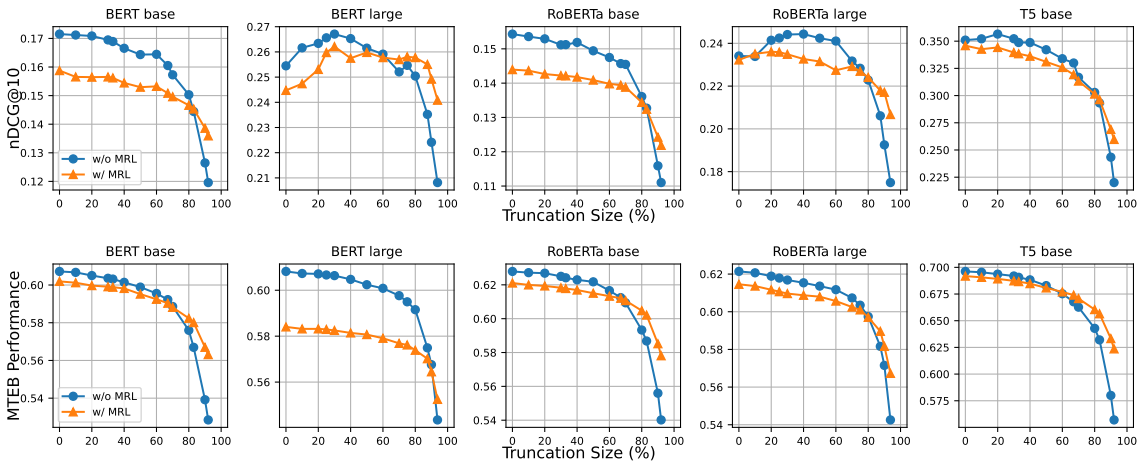

Figure 3: Performance on BEIR and MTEB benchmarks of five pairs of encoders trained with and without MRL.

validation samples. Fig. 5 in Appendix A.2 shows that all our models converge in less than 10 epochs.

Results and discussion. As before, we evaluate models on MTEB and BEIR.[2] Fig. 3 shows how downstream task performance changes with different levels of truncation for MRL and non-MRL models (see Tables 2 to 5 in the Appendix for detailed results). Surprisingly, truncated embeddings from non-MRL models outperform their MRL counterparts almost every time across all models and datasets up until about when 80% truncation is applied, at which point MRL truncation becomes increasingly more beneficial. In other words, in a controlled environment, we see that truncated vectors are more effective without MRL than with MRL, unless heavy truncation is applied. As in the experiments with publicly available state-of-the-art models made in the previous section, these results show that random truncation is often a more effective method of cost reduction than MRL, as it does not rely on any additional training cost and often outperforms truncation done on MRL models. However, MRL does become beneficial at heavy truncation levels, which are indeed common where more extreme cost reductions are desired. In fact, when plotting standard deviation across embeddings from all of our trained PLMs[3], we noted that the variance is much higher only in lower dimensions when MRL is used during training (see Fig. 4 for BERT, Fig. 6 for all other models). This suggests that MRL indeed induces more information storage in the lower dimensions of text embeddings. However, it's unclear why this is not the case for all dimensions, nor whether this can be achieved at higher levels of truncation.

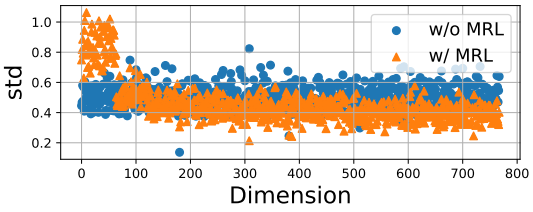

Figure 4: Standard deviation across embedding dimensions given by BERT base for different text inputs. The addition of MRL clearly increases the variance of the lower dimensions, suggesting that more information is indeed stored in lower dimensions with MRL.

4 Conclusion

We explored the effect of truncation on text embeddings obtained with and without Matryoshka Representation Learning (MRL), a training method designed to allow effective truncation. Our test with 4 open encoders, 10 newly trained text encoders and 24 embedding-based downstream tasks, show that MRL provides clear benefits at very high truncation levels, but that at lower levels is not more beneficial than applying truncation on models without MRL. In addition, we found that on a more controlled environments, truncation on non-MRL encoders is almost always superior at normal truncation levels, but that again MRL is more benefical at high truncation levels, e.g. about 80%.

[2]BEIR for smaller models BERT base and RoBERTa base, NanoBEIR for the rest.

[3]We use queries from NanoBEIR's MS Marco as inputs.

Limitations

This work has several limitations. First, our experiments only include relatively small models. While small text encoders are commonly used and recent efforts have focused on such high-performing small models, e.g. EmbeddingGemma-300M (Vera et al., 2025), state-of-the-art models are still much larger, e.g. Qwen3-Embedding-4B (Zhang et al., 2025b). Such models are not included in our study due to limited computational resources, but truncation needs to be studied on larger models as well. Second, while the training dataset used for our contrastive training is a common option in prior work, there are more recent training recipes with larger datasets to train more powerful encoders. We opted for our current choice due to limited computational resources. Finally, a larger variety of training settings could be tested, e.g. exploring the impact that different choices of MRL truncation sizes have on performance.

References

Dimitris Achlioptas. 2003. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of computer and System Sciences*, pages 671–687.

Alexander Bondarenko, Maik Fröbe, Meriem Beloucif, Lukas Gienapp, Yamen Ajjour, Alexander Panchenko, Chris Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, and 1 others. 2020. Overview of touché 2020: argument retrieval. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 11th International Conference of the CLEF Association, CLEF 2020, Thessaloniki, Greece, September 22–25, 2020, Proceedings 11*, pages 384–395. Springer.

Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A full-text learning to rank dataset for medical information retrieval. In *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38*, pages 716–722. Springer.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. 2017. Dbpedia-entity v2: A test collection for entity search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 1265–1268, New York, NY, USA. Association for Computing Machinery.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yunhsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *Preprint*, arXiv:1705.00652.

Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2553–2561.

Nathan Inkiriwang, Necva Bölücü, Garth Tarr, and Maciej Rybinski. 2025. Do we really need all those dimensions? an intrinsic evaluation framework for compressed embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 13305–13323.

William B Johnson, Joram Lindenstrauss, and 1 others. 1984. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, page 1.

Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. 2020. The multilingual Amazon reviews corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4563–4568, Online. Association for Computational Linguistics.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and 1 others. 2022. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35:30233–30249.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Markus Leippold and Thomas Diggelmann. 2020. Climate-fever: A dataset for verification of real-world climate claims. In *NeurIPS 2020 Workshop on Tackling Climate Change with Machine Learning*.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

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.

Maggie, Phil Culliton, and Wei Chen. 2020. Tweet sentiment extraction. `https://kaggle.com/competitions/tweet-sentiment-extraction`. Kaggle.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Www'18 open challenge: Financial opinion mining and question answering. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, page 1941–1942, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, page 165–172, New York, NY, USA. Association for Computing Machinery.

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

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022a. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022b. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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

Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Nan Wang, and 1 others. 2024. jina-embeddings-v3: Multilingual embeddings with task lora. *arXiv preprint arXiv:2409.10173*.

Sotaro Takeshita, Yurina Takeshita, Daniel Ruffinelli, and Simone Paolo Ponzetto. 2025. Randomly removing 50% of dimensions in text embeddings has minimal impact on retrieval and classification tasks. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27693–27714.

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.

Hayato Tsukagoshi and Ryohei Sasano. 2025. Redundancy, isotropy, and intrinsic dimensionality of prompt-based text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25915–25930, Vienna, Austria. Association for Computational Linguistics.

Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, and 1 others. 2025. Embeddinggemma: Powerful and lightweight text representations. *arXiv preprint arXiv:2509.20354*.

Ellen Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R. Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. 2021. Trec-covid: constructing a pandemic information retrieval test collection. *SIGIR Forum*, 54(1).

Ivan Vulić, Goran Glavaš, Fangyu Liu, Nigel Collier, Edoardo Maria Ponti, and Anna Korhonen. 2023. Probing cross-lingual lexical knowledge from multilingual sentence encoders. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2089–2105, Dubrovnik, Croatia. Association for Computational Linguistics.

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251, Melbourne, Australia. Association for Computational Linguistics.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916, Bangkok, Thailand. Association for Computational Linguistics.

Shuai Wang, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. 2025. 2d matryoshka training for information retrieval. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 3125–3134.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Jinsung Yoon, Rajarishi Sinha, Sercan O Arik, and Tomas Pfister. 2024. Matryoshka-adaptor: Unsupervised and supervised tuning for smaller embedding

dimensions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10318–10336.

Biao Zhang, Lixin Chen, Tong Liu, and Bo Zheng. 2025a. Smec: Rethinking matryoshka representation learning for retrieval embedding compression. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26220–26233.

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. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4):1–60.

Xiangyu Zhao, Maolin Wang, Xinjian Zhao, Jiansheng Li, Shucheng Zhou, Dawei Yin, Qing Li, Jiliang Tang, and Ruocheng Guo. 2023. Embedding in recommender systems: A survey. *arXiv preprint arXiv:2310.18608*.

A Appendix

A.1 Additional Experimental Details

	Name	Domain	Licence
Retrieval	MS MARCO (Nguyen et al., 2016)	Misc.	MIT
	TREC-COVID (Voorhees et al., 2021)	Bio-Medical	Dataset License Agreement
	NFCorpus (Boteva et al., 2016)	Bio-Medical	N/A
	FiQA-2018 (Maia et al., 2018)	Finance	N/A
	ArguAna (Wachsmuth et al., 2018)	Misc.	CC BY 4.0
	Touche-2020 (Bondarenko et al., 2020)	Misc.	CC BY 4.0
	Quora	Quora	N/A
	DBPedia (Hasibi et al., 2017)	Wikipedia	CC BY-SA 3.0
	SCIDOCS (Cohan et al., 2020)	Scientific	GNU General Public License v3.0
	FEVER (Thorne et al., 2018)	Wikipedia	CC BY-SA 3.0 l
	Climate-FEVER (Leippold and Diggelmann, 2020)	Wikipedia	N/A
	SciFact (Wadden et al., 2020)	Scientific	CC BY-NC 2.0
	Natural Questions (Kwiatkowski et al., 2019)	Scientific	CC BY-SA 3.0
	HotpotQA (Yang et al., 2018)	Scientific	CC BY-SA 4.0
Classification	AmazonCounterfactualClassification (O'Neill et al., 2021)	Reviews, Written	CC-by-4.0
	AmazonPolarityClassification (McAuley and Leskovec, 2013)	Reviews, Written	Apache 2.0
	AmazonReviewsClassification (Keung et al., 2020)	Reviews, Written	N/A
	Banking77Classification (Casanueva et al., 2020)	Written	MIT
	EmotionClassification (Saravia et al., 2018)	Social, Written	N/A
	ImdbClassification (Maas et al., 2011)	Reviews, Written	N/A
	MassiveIntentClassification (FitzGerald et al., 2023)	Spoken	Apache 2.0
	MassiveScenarioClassification (FitzGerald et al., 2023)	Spoken	Apache 2.0
	MTOPDomainClassification (Li et al., 2021)	Spoken	N/A
	MTOPIntentClassification (Li et al., 2021)	Spoken	N/A
	TweetSentimentExtractionClassification (Maggie et al., 2020)	Social, Written	N/A

Table 1: A list of datasets used in our evaluation. TREC-COVID is only available in BEIR, not in NanoBEIR.

A.2 Validation Loss Curve During Contrastive Trainings

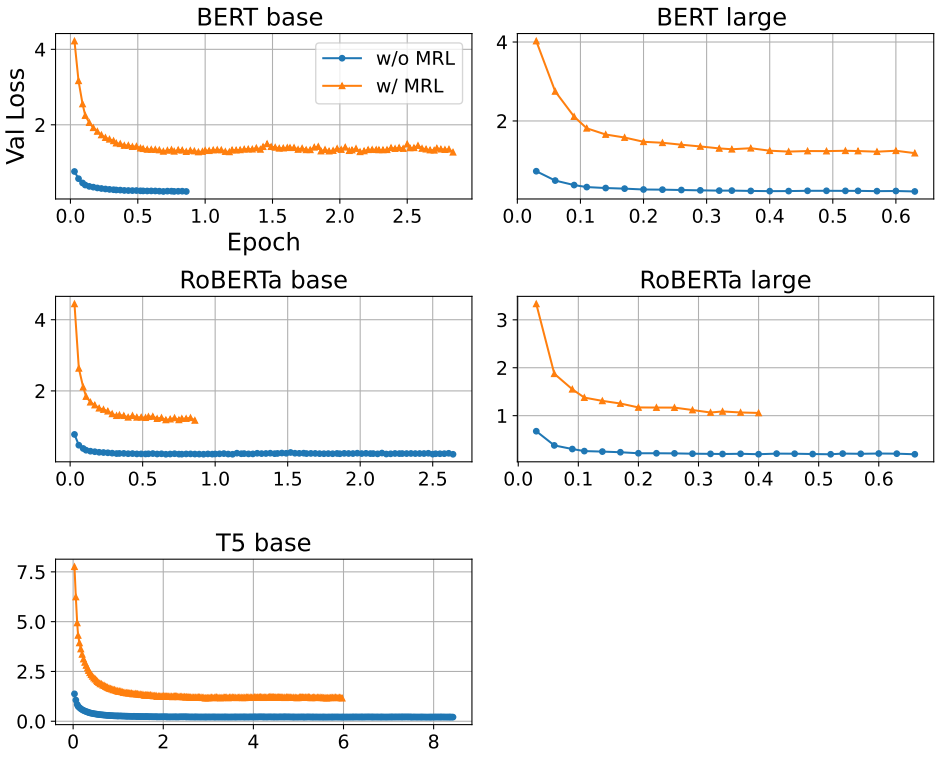

Figure 5: The validation loss curve during our contrastive learning with and without MRL for all of our five model pairs.

A.3 Performance of Text Encoders with Different Truncation Sizes

		Truncation size (%)													
Model	MRL	0	10	20	30	33	40	50	60	67	70	80	83	90	92
BERT base	No	**0.172**	**0.171**	**0.171**	**0.169**	**0.169**	**0.167**	**0.164**	**0.164**	**0.161**	**0.157**	**0.150**	0.144	0.126	0.120
	Yes	0.159	0.157	0.156	0.157	0.156	0.154	0.153	0.153	0.151	0.150	0.147	**0.145**	**0.139**	**0.136**
RoBERTa base	No	**0.154**	**0.154**	**0.153**	**0.151**	**0.151**	**0.152**	**0.149**	**0.147**	**0.146**	**0.145**	**0.136**	**0.133**	0.116	0.111
	Yes	0.144	0.144	0.143	0.142	0.142	0.142	0.141	0.140	0.139	0.139	0.134	0.132	**0.124**	**0.122**
T5 base	No	**0.351**	**0.352**	**0.357**	**0.352**	**0.349**	**0.349**	**0.342**	**0.334**	**0.330**	**0.317**	**0.303**	0.293	0.243	0.220
	Yes	0.346	0.343	0.344	0.340	0.338	0.336	0.331	0.326	0.319	0.313	0.301	**0.296**	**0.269**	**0.260**
GTR T5 large	No	0.594	0.594	0.592	0.592	0.589	0.591	0.582	0.573	0.565	0.561	0.538	0.528	0.478	0.448
EmbeddingGemma 300m	Yes	0.653	0.651	0.648	0.652	0.646	0.645	0.642	0.632	0.636	0.630	0.604	0.595	0.532	0.512

Table 2: Retrieval performance comparison between non-MRL and MRL models with different truncation sizes. All models produce embeddings with 768 dimensions. BEIR is used for BERT base and RoBERTa base, while NanoBEIR is used for the rest of models. The first three models are comparable between MRL and non-MRL trained by us while the last two are less comparable existing models. Between the comparable MRL and non-MRL pairs, scores are **bolded** when it outperforms the other.

Model	MRL	Truncation size (%)													
		0	10	20	30	33	40	50	60	67	70	80	83	90	92
BERT base	No	**0.607**	**0.607**	**0.605**	**0.604**	**0.603**	**0.601**	**0.599**	**0.596**	**0.592**	**0.589**	0.576	0.567	0.539	0.528
	Yes	0.602	0.601	0.600	0.599	0.599	0.598	0.595	0.592	0.590	0.588	**0.582**	**0.580**	**0.567**	**0.563**
RoBERTa base	No	**0.628**	**0.627**	**0.627**	**0.625**	**0.624**	**0.623**	**0.622**	**0.617**	0.612	0.609	0.593	0.587	0.556	0.540
	Yes	0.621	0.620	0.619	0.618	0.618	0.617	0.615	0.613	0.612	**0.611**	**0.605**	**0.602**	**0.585**	**0.578**
T5 base	No	**0.696**	**0.695**	**0.694**	**0.692**	**0.691**	**0.688**	**0.683**	0.675	0.668	0.663	0.643	0.632	0.580	0.557
	Yes	0.692	0.691	0.689	0.687	0.687	0.685	0.681	**0.677**	**0.674**	**0.671**	**0.660**	**0.657**	**0.633**	**0.624**
GTR T5 large	No	0.670	0.668	0.663	0.658	0.656	0.652	0.645	0.635	0.626	0.623	0.603	0.592	0.548	0.529
EmbeddingGemma 300m	Yes	0.835	0.835	0.835	0.835	0.835	0.834	0.833	0.831	0.831	0.829	0.825	0.822	0.806	0.801

Table 3: Classification performance comparison between non-MRL and MRL models with different truncation sizes. All models produce embeddings with 768 dimensions. The first three models are comparable between MRL and non-MRL trained by us while the last two are less comparable existing models. Between the comparable MRL and non-MRL pairs, scores are **bolded** when it outperforms the other.

Model	MRL	Truncation size (%)													
		0	10	20	25	30	40	50	60	70	75	80	88	90	94
BERT large	No	**0.254**	**0.262**	**0.263**	**0.266**	**0.267**	**0.265**	**0.261**	**0.259**	0.252	0.255	0.250	0.235	0.224	0.208
	Yes	0.245	0.247	0.253	0.260	0.262	0.257	0.260	0.258	**0.257**	**0.258**	**0.258**	**0.255**	**0.249**	**0.241**
RoBERTa large	No	**0.234**	0.234	**0.241**	**0.242**	**0.244**	**0.244**	**0.242**	**0.241**	**0.232**	**0.228**	0.223	0.206	0.193	0.175
	Yes	0.232	**0.235**	0.236	0.236	0.235	0.233	0.231	0.227	0.229	0.227	**0.224**	**0.218**	**0.217**	**0.207**
E5 large	No	0.600	0.598	0.598	0.597	0.595	0.594	0.594	0.586	0.576	0.572	0.560	0.517	0.498	0.427
Qwen 3 0.6B	Yes	0.582	0.578	0.578	0.578	0.580	0.577	0.571	0.567	0.555	0.549	0.550	0.530	0.515	0.475

Table 4: Retrieval performance comparison between non-MRL and MRL models with different truncation sizes. All models produce embeddings with 1024 dimensions. BEIR is used for BERT base and RoBERTa base, while NanoBEIR is used for the rest of models. The first three models are comparable between MRL and non-MRL trained by us while the last two are less comparable existing models. Between the comparable MRL and non-MRL pairs, scores are **bolded** when it outperforms the other.

Model	MRL	Truncation size (%)													
		0	10	20	25	30	40	50	60	70	75	80	88	90	94
BERT large	No	**0.608**	**0.607**	**0.607**	**0.607**	**0.606**	**0.605**	**0.602**	**0.601**	**0.598**	**0.595**	**0.592**	**0.575**	**0.568**	0.544
	Yes	0.584	0.583	0.583	0.583	0.582	0.581	0.581	0.579	0.577	0.576	0.574	0.570	0.564	**0.552**
RoBERTa large	No	**0.621**	**0.621**	**0.619**	**0.618**	**0.617**	**0.615**	**0.614**	**0.612**	**0.607**	**0.603**	0.597	0.582	0.572	0.543
	Yes	0.615	0.614	0.612	0.611	0.610	0.609	0.608	0.606	0.602	0.601	0.597	**0.590**	**0.582**	**0.567**
E5 large	No	0.687	0.685	0.682	0.681	0.680	0.678	0.673	0.666	0.658	0.652	0.644	0.614	0.597	0.567
Qwen 3 0.6B	Yes	0.704	0.703	0.702	0.701	0.700	0.699	0.693	0.688	0.671	0.670	0.665	0.648	0.636	0.608

Table 5: Classification performance comparison between non-MRL and MRL models with different truncation sizes. All models produce embeddings with 1024 dimensions. The first three models are comparable between MRL and non-MRL trained by us while the last two are less comparable existing models. Between the comparable MRL and non-MRL pairs, scores are **bolded** when it outperforms the other.

A.4 Standard Deviations of Each Dimension

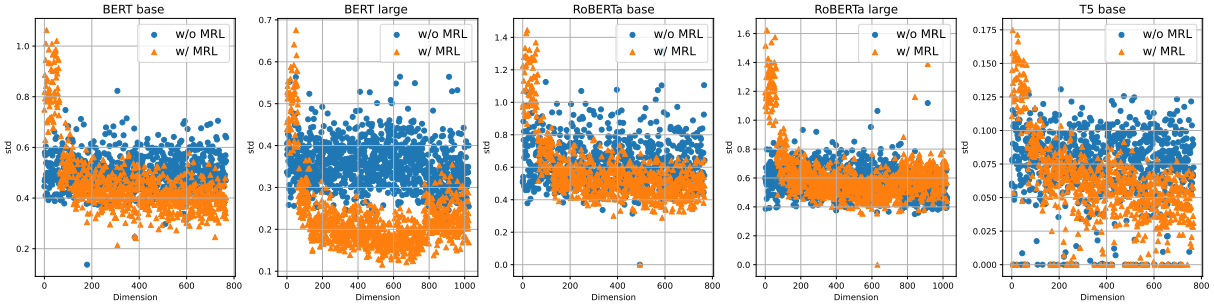

Figure 6: Standard deviations of values taken by each dimension when encoding different texts.