Randomly Removing 50% of Dimensions in Text Embeddings has Minimal Impact on Retrieval and Classification Tasks

Sotaro Takeshita[1], Yurina Takeshita[2], Daniel Ruffinelli[1], Simone Paolo Ponzetto[1]
[1]Data and Web Science Group, University of Mannheim, Germany
[2]Independent researcher
{sotaro.takeshita, druffinelli, ponzetto}@uni-mannheim.de

Abstract

In this paper, we study the surprising impact that truncating text embeddings has on downstream performance. We consistently observe across 6 state-of-the-art text encoders and 26 downstream tasks, that randomly removing up to 50% of embedding dimensions results in only a minor drop in performance, less than 10%, in retrieval and classification tasks. Given the benefits of using smaller-sized embeddings, as well as the potential insights about text encoding, we study this phenomenon and find that, contrary to what is suggested in prior work, this is not the result of an ineffective use of representation space. Instead, we find that a large number of uniformly distributed dimensions actually cause an increase in performance when removed. This would explain why, on average, removing a large number of embedding dimensions results in a marginal drop in performance. We make similar observations when truncating the embeddings used by large language models to make next-token predictions on generative tasks, suggesting that this phenomenon is not isolated to classification or retrieval tasks. Our code is attached to the submission.[1]

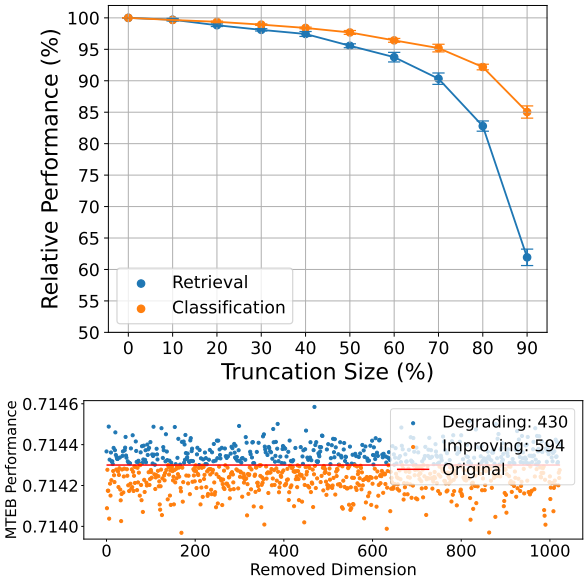


Figure 1: (top) Regardless of the selection, removing 50% of embedding dimensions results in less than 5% performance drop. (bottom) This seems related to is due to many dimensions that lower performance (depicted in blue).

1 Introduction

As text embeddings are used in various applications such as retrieval augmented generation (Li et al., 2025), question answering (Karpukhin et al., 2020), or text retrieval (Liu et al., 2021), there have been extensive research efforts not only aiming at improving their performance but also to understand them. A number of works explore *what* text embeddings encode, such as using probing methods in well-controlled setups to check the information encoded by embeddings (Hewitt and Manning, 2019; Kulmizev et al., 2020). However, less has been explored on *how* information is encoded. Existing works in this direction often assess isotropy (or anisotropy) in text embeddings, that is, whether text embeddings are scattered (or concentrated) in representation space (Ait-Saada and Nadif, 2023; Godey et al., 2024). While these works provide insights into geometric properties of text embeddings, they often focus less on the impact that these properties have on downstream tasks.

To fill this gap, we look at how text embeddings use the representation space *through the lens of their impact on downstream task performance.* In our first experiment, we examine how well text encoders use the embedding space by measuring performance when removing K% of embedding dimensions. Through extensive testing with 6 text encoders, including a large language model (LLM), and 26 embedding-based tasks (e.g., passage retrieval and intent classification), we find that, surprisingly, the resulting embeddings still achieve

[1]https://sotaro.io/papers/trunbed

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 27705–27726
November 4-9, 2025

comparable performance to the original embeddings even when more than half of the dimensions are removed. Regardless of the selection of removed dimensions, reduced embeddings can retain 95% and 90% of the original performance, in classification and retrieval tasks, respectively (Fig. 1: top). By measuring Spearman correlation, we find that indeed, truncation seems to preserve the properties of the space well enough that the rankings used for retrieval and classification are mostly unaffected. These results suggest an inefficient use of the representation space by text encoders.

To study whether embeddings use the representation space effectively, we integrate three well-studied concepts into our study. Specifically, we test if the cause is (i) embeddings gathering in a narrow cone in representation space (Xiao et al., 2023), (ii) redundancy in dimensions (Jing et al., 2021), or (iii) a few outlier dimensions that determine performance (Kovaleva et al., 2021). While all of these properties are present in every models, we do not find strong relations to our initial observation, calling for a new perspective.

To this end, we take inspiration from input attribution methods (Sundararajan et al., 2017; Bastings et al., 2022), and analyze the contribution of each embedding dimension on downstream performance. We find that every model contains many dimensions that negatively impact performance, dubbed degrading dimensions. For instance, we identify 430 degrading dimensions (out of 1024) in E5-large (Wang et al., 2022) (Fig. 1: bottom). The degrading dimensions are uniformly distributed across embedding features. This suggests a possible reason for our initial observation, that is, when we remove dimensions randomly, both the positively and negatively contributing features are removed, resulting in a marginal performance drop. Indeed, when removing only the degrading dimensions, the performance drop is much slower compared to removing random dimensions, or the performance improves from the original embeddings. We also find that a significant number of degrading dimensions are shared across tasks, indicating the potential for further improvements in text encoders.

While our focus is on embeddings produced by text encoders, we also find similar results when truncating the embeddings used by LLMs for next-token prediction in text generation tasks, i.e., tasks where these truncated embeddings are repeatedly used. However, the results in this case are more task-dependent, as we find that in some tasks, performance is severely degraded.

Our contributions are the following:

- We consistently find across several models and downstream tasks, that text embeddings retain more than 90% of their original performance even after randomly removing 50% of their dimensions. We make similar but less consistent findings about the embeddings used by causal language models.
- We identify that the representations obtained from state-of-the-art text embedders can contain a significant amount of dimensions that have a negative impact on many downstream tasks, but more research is needed to understand their role in text representations and to possibly improve existing text encoders.

2 Impact of Embedding Truncation

In this section, we study how well text embeddings use their dimensions by considering a simple hypothesis: if text encoders are effectively using embedding dimensions, removing some features in an arbitrary manner should have a noticeably negative effect on performance. To test this hypothesis, we look at the impact that different methods of embedding truncation have on downstream performance.

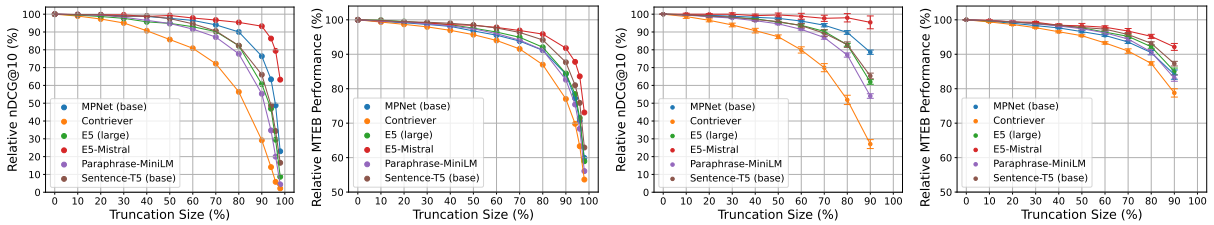
2.1 Experimental Settings

Models. We consider 6 state-of-the-art models, including an LLM-based model, with various sizes and training configurations. All models are contrastively trained after pre-training (see Table 5 for a list of models).

Tasks. For downstream task evaluation, we take 14 retrieval and 12 classification datasets from BEIR (Thakur et al., 2021)[2] and MTEB (Muennighoff et al., 2023) benchmarks (see Table 6 for a list of datasets).

Truncation methods. We evaluate with two different truncation approaches: (i) last K% truncation: we simply remove the last K% of dimensions from embeddings, and (ii) random K% truncation: we uniformly sample K% of the features to be removed. We repeat this process ten different times and report standard deviation in Fig. 2c and 2d.

[2]Due to its high computational demand, for the E5-Mistral model, we use NanoBEIR, which is a subset of the BEIR.



(a) Last K%, BEIR (b) Last K%, MTEB (c) Random K%, BEIR (d) Random K%, MTEB

Figure 2: Relative performance when (a, b) last and (c, d) random K% of dimensions are removed. Error bars in (c, d) are drawn from the results of ten different random removals. The results per dataset are shown in Fig. 9, 10.

2.2 Results and Discussion

Last K% truncation. Fig. 2a and 2b show the results on each benchmark. Relative performance only drops to below 80% when 80% of the dimensions are removed for five out of six models. In an extreme case, E5-Mistral retains 90% of its original performance even when 90% of its dimensions are removed. This suggests that most of the features in these text embeddings do not have a big impact on downstream performance. We further compare the two rankings of retrieved documents produced by original embeddings and truncated embeddings, and indeed see that Spearman's rank-order correlation remains high even after truncating 50%, higher than 0.8 for all models (details in Fig. 8).

Random K% truncation. Fig. 2c and 2d show a very similar pattern in performance reduction curves compared to our previous experiments, and the standard deviations of relative performance between ten random runs are small. This means that regardless of the selection of removed dimensions, models are able to retain most of their downstream performance. In the next section, we look at existing theories in the literature that may explain this phenomenon.

3 Effective Use of Representation Space

In this section, we investigate why text embeddings can be significantly reduced in size without much performance loss. We do this from the perspective of three different concepts from prior works that explore how embeddings (in)effectively use the representation space: anisotropy, dimensional collapse, and outlier dimensions.

3.1 Anisotropy in Embeddings

A number of existing works report that neural network-based encoders, not only for texts (Rudman et al., 2022; Hämmerl et al., 2023; Godey et al., 2024) but also for images (Wang and Isola, 2020), tend to produce anisotropic embeddings, meaning that the encoders map different input data points into a narrow cone without fully exploiting the representation space. The two main characteristics of anisotropic embeddings are (i) distorted variance in values taken by different dimensions and (ii) high correlations between different dimensions (i.e., features) (Rudman and Eickhoff, 2023). As this property may explain our earlier observations, we explore if anisotropy in text embeddings can be a predictor of performance with truncated embeddings.

Experimental setup. Prior work has shown that contrastively training models makes their embeddings less anisotropic, which in turn improves downstream performance (Ni et al., 2022; Chen et al., 2023). So, to obtain embeddings with different levels of anisotropy, we contrastively train two pre-trained models, BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020), storing intermediate checkpoints along the way. The expectation is that, as anisotropy decreases with more training, performance should increase for full-size embeddings, but decrease for truncated embeddings, as less anisotropy means the model is making more effective use of the representation space. We train the models with a mixture of SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018) as training and validation data (Reimers and Gurevych, 2019; Gao et al., 2021), and use InfoNCE as a loss function (Oord et al., 2019). The training is terminated upon convergence of the validation loss. To measure anisotropy, we use two common metrics: uniform loss (Wang and Isola, 2020; Ni et al., 2022) and IsoScore (Rudman et al., 2022), the former decreases and the latter increases its values as the target embeddings become less anisotropic. We apply the last K% truncation to obtain reduced embeddings, and use 13 datasets from NanoBEIR

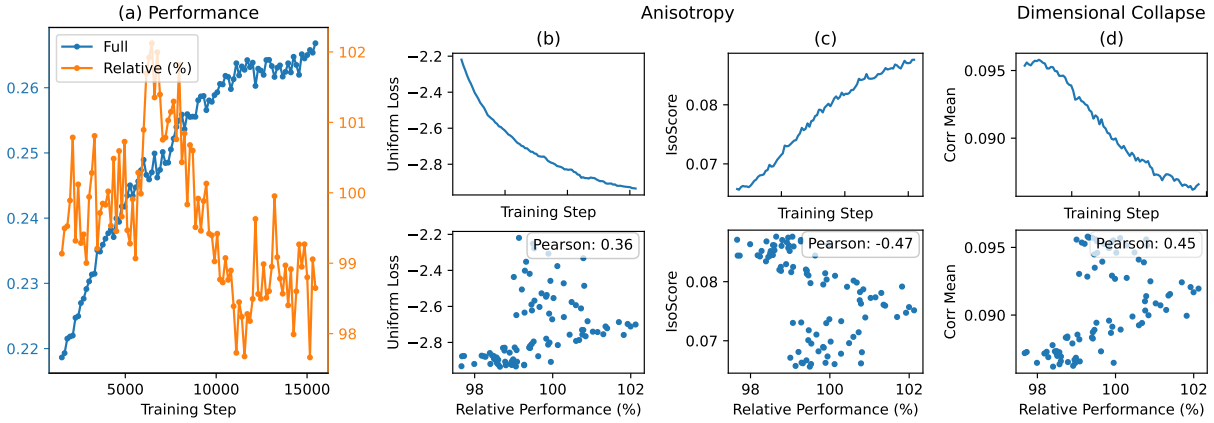


Figure 3: As a result of contrastive learning for T5, downstream task performance increases (a: Full), and the use of embedding space measured through Uniform Loss (↓) and IsoScore (↑) for anisotropy (b, c: top) and Corr Mean (↓) for dimensional collapse (d: top) also improves. However, the relative performance does not change over the training (a: Relative), therefore, there is no strong correlation between relative performance and representation quality measures (b, c, d: bottom).

to evaluate the final performance.

Results. Fig. 3a shows how the full-sized embeddings improve performance during the training together with the decrease in anisotropy as shown in Fig. 3b, c (top) for T5 (result for BERT is shown in Fig. 11). As expected, more training results in increased downstream performance and decreased anisotropy. However, the relative performance achieved by half-sized embeddings does not change over training steps. Fig. 3b, c (bottom) shows the relation between the relative performance and two anisotropy measures. We do not see correlating patterns because even though different checkpoints have different degrees of anisotropy, their relative performance is quite stable. We also compute the Pearson correlation between relative performance and the two anisotropy metrics, but do not observe any strong correlation (0.36 for uniform loss, -0.47 for IsoScore). These results indicate that even when models make better use of the representation space, as measured by anisotropy, truncated embeddings still result in marginal performance drops.

3.2 Dimensional Collapse

Existing works, especially in the computer vision community, report that the representations produced by neural network-based encoders have certain dimensions collapsed, and use only a lower-dimensional subspace (Huang et al., 2023; Hua et al., 2021; He and Ozay, 2022). While this is conceptually similar to anisotropy, dimensional collapse focuses on the correlation between dimensions. In this section, we explore the relation of dimensional collapse in text embeddings to the high relative performance we observe, as the embeddings with highly correlated dimensions can be robust to feature removal.

Experimental setup. We follow Hua et al. (2021) and use the mean of the correlation coefficient between dimensions, computed over embeddings obtained by encoding 10K English paragraphs from Wikipedia[3]. Same as our experiments on anisotropy in §3.1, we use the contrastively trained checkpoints of two model families and NanoBEIR evaluation. We hypothesize that the impact of dimension truncation can remain low for the models where the dimensions are highly correlating.

Results. Fig. 3d (top) shows that the contrastive training reduces correlations between dimensions for T5 (result for BERT is shown in Fig. 11 in the Appendix). As shown in existing works (Jing et al., 2021; He and Ozay, 2022), downstream task performance improves as the correlations weaken. However, as shown in Fig. 3d (bottom), we do not observe any relation between feature correlations and how performance drops after truncating the last 50% of dimensions; the Pearson correlation between them is 0.45. Given these results, we conclude that dimensional collapse, i.e. correlation between dimensions, does not explain the success of truncated embeddings.

[3] sentence-transformers/simple-wiki

	# of Outliers	Outlier	Non-outlier
MPNet (base)	5	0.576	0.576 ± 9e-04
Contriever	3	0.525	0.524 ± 1e-03
E5 (large)	1	0.577	0.577 ± 4e-04
E5-Mistral	32	0.651	0.626 ± 6e-04
Para-MiniLM	2	0.483	0.483 ± 6e-04
ST5 (base)	2	0.489	0.489 ± 9e-04

Table 1: Effect of removing outlier dimensions on downstream task performance. As a comparison, we also remove the same number of ten different non-outlier dimensions and report the average and standard deviation of achieved performance.

3.3 Outlier Dimensions in Embeddings

Several papers report that there are a few dimensions in the weights of pre-trained language models (PLMs) that take abnormally high or low values, known as outlier dimensions (Kovaleva et al., 2021; Puccetti et al., 2022; Hämmerl et al., 2023). PLM weights are often redundant and can be removed without much performance loss (Michel et al., 2019; Bian et al., 2021); however, Kovaleva et al. (2021) show that removing such outlier dimensions from models, even though there are only a few, can massively reduce performance. A follow-up work by Hämmerl et al. (2023) reports that a similar trend exists in multilingual models as well. While they study outlier dimensions in model weights, in this paper, we extend this concept to identify outlier dimensions in text embeddings and explore their interactions with our interest, the high relative performance.

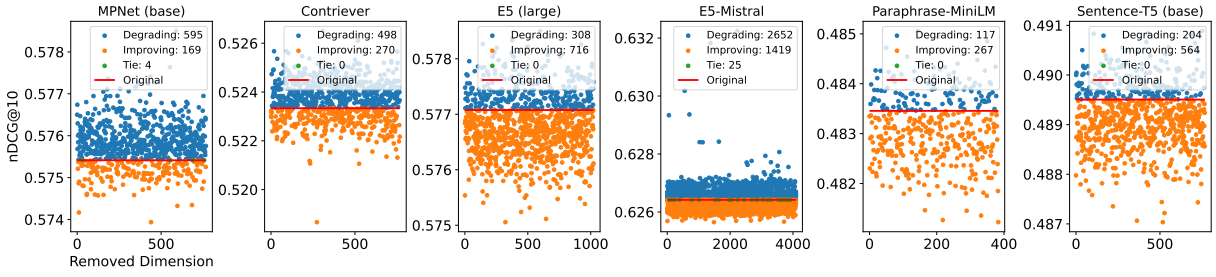
Experimental setup. First, we aim to identify outlier dimensions within text embeddings produced by text encoders. To this end, we examine the embedding obtained by averaging the embeddings of all the query texts from the NanoBEIR datasets. We follow the definition of outlier dimensions used by Kovaleva et al. (2021), that is, the dimensions that deviate more than 3σ from the standard deviation of all values in the average embedding. After identifying them, we assess their effects on downstream tasks by comparing the performance achieved by: (i) the embeddings without outlier dimensions, and (ii) the embeddings without non-outlier dimensions, the same number as outlier dimensions. The second configuration is our control trial. We experiment by removing ten different sets of non-outlier dimensions.

Results. The number of outlier dimensions for each model and their effect on performance are shown in Table 1. The number of outlier dimensions changes with different models; however, it remains low in all cases. This is similar to the outliers within weights reported by Kovaleva et al. (2021). The highest proportion of outlier dimensions to the full embedding is observed with E5-Mistral, that is 0.8% (32 out of 4096 dimensions). The figure also shows that, for five out of six models, the effect of removing outlier dimensions is not beyond the ones where we remove non-outlier dimensions, indicating that, outlier dimensions do not play a critical role. The exception is E5-Mistral, where removing outlier dimensions results in a slight increase in performance with regard to non-outlier removal counterparts, but the gap to the average performance non-outlier removal runs is too small, 0.025 in nDCG@10, to explain our initial observation. These observations are strong indications that the outlier dimensions are unlikely to be the reasons for the low drop rate, which is our interest in this paper.

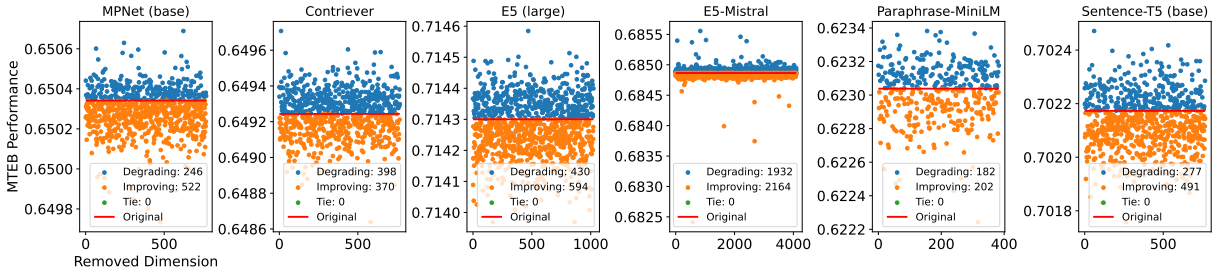
4 Dimension Attribution Analysis

Our previous experiments suggest that truncated embeddings perform well on downstream tasks, even when they seem to make good use of the representation space through existing concepts, e.g. when embeddings are better spread across the representation space, or when there is less correlation between features, or when outlying dimensions are considered. Since this implies that whatever dimensions are left after truncation are still useful, in this section, we study the impact that each dimension has on downstream performance.

Method. We take inspiration from existing works on input attribution, a family of methods that rely on perturbing inputs to determine feature importance to explain model predictions (Sattarzadeh et al., 2021; Wu et al., 2023). Specifically, we repeatedly evaluate performance by disabling one dimension at a time. This enables us to measure the contribution of each dimension to the downstream task performance in isolation. Zeroing model weights is the common way to analyze their role in model behavior (Serrano and Smith, 2019; Zhang et al., 2024b). However, as we focus on dimensions only in embeddings, we simply remove the target dimension from the embeddings. This is a preferred approach because, as each dimension can take a wide variety of values, zeroing values may have varying impacts on final performance.



(a) NanoBEIR



(b) MTEB

Figure 4: Average performance on all datasets in the NanoBEIR and MTEB benchmarks after removing each dimension in the input embeddings. The red horizontal line indicates the performance achieved by the original embedding, and each point is the performance without the corresponding dimension. Blue points indicate that they are negatively impacting the performance as they are above the red line.

Experiments setup. We take 13 retrieval and 12 classification datasets from NanoBEIR and MTEB, respectively, and perform our analysis with the same six models used in Section 2, 3.

Main results. Fig. 4 shows the results on NanoBEIR and MTEB. Each point indicates the downstream task performance achieved without the corresponding dimension. The red line is the original performance achieved by the full-sized embeddings. Points are highlighted in blue (or orange) when they are better (or worse) than the original performance. There are two main observations in the figures. (i) In all model-benchmark combinations, a surprisingly large number of dimensions improve the performance when they are removed (blue in the figures). In other words, there are a large number of dimensions, more than half in some cases, that are degrading the embeddings' performance. In the remainder of the paper, we call them degrading dimensions. For instance, we find 498 and 398 degrading dimensions in embeddings produced by Contriever on NanoBEIR and MTEB, that is, 65% and 52% of the whole embeddings. (ii) The degrading dimensions are uniformly distributed across embedding dimensions without clustering in some areas.

These observations provide an explanation for why removing a large number of dimensions has minimal impact on performance. Since there are many degrading dimensions spread across embedding dimensions, random dimension removal leads to removing dimensions that both improve and degrade performance, resulting in a marginal performance drop as a whole. We further conduct an experiment where we only remove the degrading dimensions. As the results shown in Fig. 5 (blue plot), we observe that the relative performance

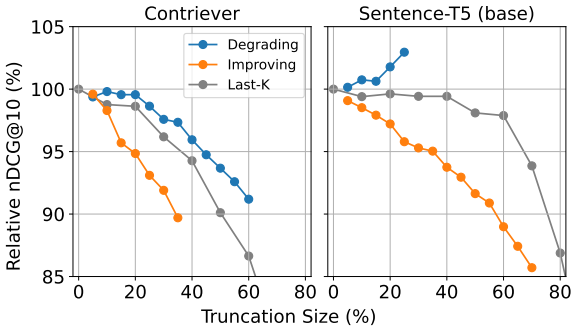


Figure 5: As we remove the degrading dimensions (blue plot), the relative performance for Sentence-T5 (figure on right) improves over the original embeddings. For Contriever (figure on left), while we do not see the improvements, however, the decay is slower than the last-k truncation. On the other hand, when only the improving dimensions are removed (orange plot), the performance decreases rapidly for both models. Results for other models are shown in Fig. 12.

keeps improving for some models (e.g., Sentence-T5), and even for the models whose truncated embeddings drop their performance (e.g., Contriever), the speed of decay is slower than random truncation. Conversely, when we remove only the dimensions that are improving the performance (dimensions in orange in Fig. 4a), the performance drops rapidly compared to the random truncation (Fig. 5: orange plot).

These results indicate that there is a new aspect of improvements in current text embedding models, as many of the dimensions are damaging performance. Future work can explore training objectives or model architectures that can reduce degrading dimensions to boost the model performance, similarly to how the training objective proposed by Jing et al. (2021) mitigates dimensional collapse or the novel Transformer architecture introduced by He et al. (2024) that can reduce the number of outlier dimensions.

Outlier degrading dimensions in E5-Mistral. Some degrading dimensions in E5-Mistral's embeddings have higher impacts compared to other models when evaluated on NanoBEIR (Fig. 4a). As we observe the outlier dimensions in E5-Mistral also have stronger impacts than other models (§3.3), we draw a connection from the outlier dimensions to the degrading dimensions. As shown in Fig. 13, there are 19 degrading dimensions that have stronger negative impacts than the other degrading dimensions (ODD: Outlier degrading dimensions), i.e., performance degraded at least 3σ from the mean of the performance, and 12 of them also take outlying values (ODD $\cap$ OD in the figure). This explains our earlier observation in §3.3: not all outlier dimensions (OD) have a strong impact on downstream task performance, however, some indeed have outlying impacts. We speculate that the model's extremely high relative performance (e.g., 90% relative performance when 90% of the dimensions are truncated) may be due to these outlier degrading dimensions.

Shared degrading dimensions. We test if the same dimension appears as a degrading one in multiple datasets. To this end, we independently identify degrading dimensions for each dataset in NanoBEIR and count the ones shared in multiple datasets. Fig. 6 shows the result. We can see that while there is no degrading dimension shared across all 13 datasets in any models, they are indeed commonly degraded dimensions across some

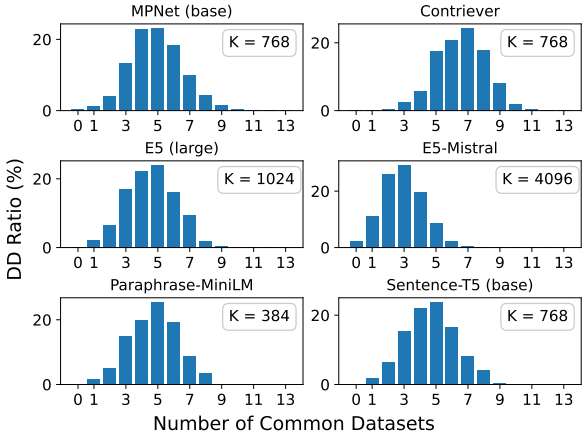


Figure 6: The ratio of degrading dimensions to the size of original embedding sizes (K in the figure) that are shared across a certain number of datasets.

	BEIR		MTEB	
	Trun. (%)	PCA (%)	Trun. (%)	PCA (%)
MP	97.7	99.4	96.6	99.6
Cont	87.3	77.3	95.4	98.5
E5-L	95.9	90.6	97.7	99.8
E5-M*	99.6	100.6	98.2	100.4
Para	94.6	99.3	97.4	99.5
ST5	95.9	100.2	97.8	99.9

Table 2: Comparison of relative performance between random truncation (Trun.) and PCA when reducing embedding size by 50%. We use NanoBEIR for E5-Mistral.

datasets, i.e., MPNet's more than 20% of dimensions degrade performance in five datasets. This observation hints at a possibility to explore domain-specific methods to identify degrading dimensions, which is left for future work.

Random Truncation vs PCA. We explore the practical value of embedding truncation by comparing it to a popular dimension reduction method: PCA (Raunak et al., 2019; Zhang et al., 2024a). To this end, we compare the performance of embeddings that are halved by the PCA and the average of ten different runs of random truncation (see §2). We use the implementation from scikit-learn (Pedregosa et al., 2011), and use 20k sentences from the all-nli dataset for training[4]. Table 2 shows the results on BEIR (NanoBEIR for E5-Mistral) and MTEB. While random truncation does not require any training or additional computation at inference time, their relative performances are surprisingly close to PCA, even outperforming it in some cases,

[4] sentence-transformers/all-nli

Model	Dataset	Method	Perf (Relative)
Llama	MMLU (acc)	Full	0.681 (1.000)
		First	0.580 (**0.852**)
		Last	0.586 (**0.861**)
	SQUAD-V2 (best exact)	Full	51.87 (1.000)
		First	50.07 (**0.965**)
		Last	50.07 (**0.965**)
	GSM8K (exact match (strict))	Full	0.764 (1.000)
		First	0.009 (0.012)
		Last	0.014 (0.018)
Qwen	MMLU (acc)	Full	0.718 (1.000)
		First	0.709 (**0.988**)
		Last	0.709 (**0.987**)
	SQUAD-V2 (best exact)	Full	50.12 (1.000)
		First	50.07 (**0.999**)
		Last	50.07 (**0.999**)
	GSM8K (exact match (strict))	Full	0.766 (1.000)
		First	0.045 (0.059)
		Last	0.011 (0.014)

Table 3: Performance on three benchmark datasets when the last hidden representations and the unembedding layer are reduced by half. Relative performance is bolded when it reaches 80% of the original performance. The results on the rest of the datasets are shown in Table 4 in the Appendix.

e.g., Contriever on BEIR. This result exhibits random truncation as an extremely simple and cheap approach to reduce text embedding's dimensionality.

Truncated Representations in Causal Language Models. While the focus of this paper is on embeddings produced by text encoders, in this section, we study the impact that embedding truncation has on causal language modeling.

To this end, we consider two LLMs, Llama 3.1 8B (Grattafiori et al., 2024) and Qwen 2.5 7B (Qwen et al., 2025), and evaluate on various six tasks after removing half of the last hidden representations before they are projected to the vocabulary space (Hendrycks et al., 2020; Rajpurkar et al., 2018; Dua et al., 2019; Gordon et al., 2012; Cobbe et al., 2021; Zellers et al., 2019). We test removing the first and last half of the representations, and reduce the unembedding matrix correspondingly. We use Language Model Evaluation Harness (Gao et al., 2024) for our evaluation.

Table 3, 4 shows the results. On three out of six tasks, both models retain more than 80% of the original performance, and in these cases, similarly to our embedding-based experiments, how to reduce representations (removing first or last) does not have an impact, indicating the presence of inefficient representation space usage by LLMs. However, contrary to our embedding-based evaluation results, the high relative performance is not observed in all datasets, e.g., on GSM8K, the original performance is heavily lost with all models and reducing methods, leaving a dedicated study on LLMs for our future studies.

5 Related Works

Inefficient use of representation space within the weights of PLMs and LLMs is hinted at by a number of papers showing these models are robust to pruning (Michel et al., 2019; Budhraja et al., 2020; Chen et al., 2021; Zheng et al., 2022). In the context of text embeddings, several studies analyze their geometric properties, such as anisotropy (Hämmerl et al., 2023; Godey et al., 2024; Razzhigaev et al., 2024). However, their influence on downstream tasks is under exploration. Ait-Saada and Nadif (2023) show a limited influence of anisotropy on text clustering. Our work adds more evidence of such limited influence of anisotropy on 26 embedding-based tasks.

The work closest to ours is Kovaleva et al. (2021) in which the authors identify a few dimensions within BERT's weights that are more impactful than the others. Its successor works analyze outlier dimension in multilingual models (Hämmerl et al., 2023), analyze their properties (Rudman et al., 2023), or identify similar dimensions in LLMs (He et al., 2024). Differently, in this paper, we focus on properties of embedding dimensions instead of model weights, and while we observe outlier dimensions in embeddings, we show that they do not have a strong influence on task performance compared to the ones in model weights.

A concurrent work by Tsukagoshi and Sasano (2025) shows how little dimension truncations impact performance with prompt-based text encoders. They approach this observation from a perspective of redundancy in embeddings. They take two concepts, namely isotropy and intrinsic dimensionality, and show that the models with higher redundancy in produced embeddings are more robust to truncation. In addition, they show that prompt design has an influence on this phenomenon. Our work complements this work by (i) conducting experiments with a more diverse set of models, including non-

prompt-based encoders and LLM text generators, (ii) conducting additional controlled experiments, such as the use of a continuous set of contrastively trained models to analyze redundancy (anisotropy and dimensional collapse) which allows us to have more comparable models, and finally (iii) shedding light on a new perspective to analyze the model's use of embeddings space, namely dimensional attribution analysis.

6 Conclusion

In this paper, we explored a surprisingly small effect of randomly removing dimensions from text embeddings on downstream task performance. We showed that 6 text encoders can retain 90% of the original performance even when 50% of the dimensions are removed, consistently for 26 embedding-based downstream tasks. Through a series of analyses, we identified a significant number of dimensions in text embeddings that are lowering downstream performance, distributed across embeddings, which would explain our initial observation. We also observed a similar effect during the text generation by causal language models in some cases.

Limitations

This work has the following limitations: (i) The cause of the degrading dimensions is still unknown. While we find that there is a large number of dimensions in text embeddings that lower the downstream task performance, this work does not explore how they emerge. For instance, identifying in which stage of model construction (e.g., early stage of pre-training) such degrading dimensions start to appear remains as future work for us. (ii) We focused on removing one dimension at a time in our dimension attribution analysis experiments (§4) without taking combinations of multiple dimensions into account. While such complex analysis may provide us with more insights about degrading dimensions, this would require an extremely high computational resource, preventing us from exploring this direction. (iii) Our experiments only cover models trained for the English language and datasets that are in English. While we confirm our findings through extensive experiments, the language is limited to English, without covering other languages, leaving a possibility of different behaviours in non-English languages. (iv) The relation to models trained with the Matryoshka Representation Learning (MRL) method (Kusupati et al., 2022) is not explored. The MRL models are trained with a tailored objective function so that their output representations can be truncated without large performance loss. In our preliminary experiment, we compared two MPNet variants trained with standard and MRL objectives, and observed that the performance of truncated embeddings from both models was comparable; also, the change in relative performance with different degrees of truncation was comparable to non-MRL encoders. While larger-scale experiments to assess the impact of MRL on text encoders are interesting, such experiments require powerful computational infrastructure with large GPU memories, therefore, we are unable to pursue this direction.

Acknowledgments

The work presented in this paper is funded by the German Research Foundation (DFG) under the VADIS (PO 1900/5-1; EC 477/7-1) project, and also supported by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant INST 35/1597-1 FUGG.

References

Mira Ait-Saada and Mohamed Nadif. 2023. Is Anisotropy Truly Harmful? A Case Study on Text Clustering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1194–1203, Toronto, Canada. Association for Computational Linguistics.

Jasmijn Bastings, Sebastian Ebert, Polina Zablotskaia, Anders Sandholm, and Katja Filippova. 2022. "Will You Find These Shortcuts?" A Protocol for Evaluating the Faithfulness of Input Salience Methods for Text Classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 976–991, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuchen Bian, Jiaji Huang, Xingyu Cai, Jiahong Yuan, and Kenneth Church. 2021. On Attention Redundancy: A Comprehensive Study. pages 930–945.

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.

Aakriti Budhraja, Madhura Pande, Preksha Nema, Pratyush Kumar, and Mitesh M. Khapra. 2020. On the weak link between importance and prunability of attention heads. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3230–3235, Online. 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.

Qian Chen, Wen Wang, Qinglin Zhang, Siqi Zheng, Chong Deng, Hai Yu, Jiaqing Liu, Yukun Ma, and Chong Zhang. 2023. Ditto: A Simple and Efficient Approach to Improve Sentence Embeddings. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5868–5875, Singapore. Association for Computational Linguistics.

Xiaohan Chen, Yu Cheng, Shuohang Wang, Zhe Gan, Zhangyang Wang, and Jingjing Liu. 2021. Early-BERT: Efficient BERT training via Early-bird lottery tickets. Stroudsburg, PA, USA. Association for Computational Linguistics.

cjadams, Daniel Borkan, inversion, Jeffrey Sorensen, Lucas Dixon, Lucy Vasserman, and nithum. 2019. Jigsaw unintended bias in toxicity classification. https://kaggle.com/competitions/jigsaw-unintended[...]. Kaggle.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint*. ArXiv:2110.14168 [cs].

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers. Stroudsburg, PA, USA. 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.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. 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 2368–2378, 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.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

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.

Nathan Godey, Éric Clergerie, and Benoît Sagot. 2024. Anisotropy Is Inherent to Self-Attention in Transformers. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 35–48, St. Julian's, Malta. Association for Computational Linguistics.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. SemEval-2012 Task 7: Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference*

and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), pages 394–398, Montréal, Canada. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models. *arXiv preprint*. ArXiv:2407.21783 [cs].

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.

Bobby He, Lorenzo Noci, Daniele Paliotta, Imanol Schlag, and Thomas Hofmann. 2024. Understanding and Minimising Outlier Features in Transformer Training. *Advances in Neural Information Processing Systems*, 37:83786–83846.

Bobby He and Mete Ozay. 2022. Exploring the Gap between Collapsed & Whitened Features in Self-Supervised Learning. In *Proceedings of the 39th International Conference on Machine Learning*, pages 8613–8634. PMLR. ISSN: 2640-3498.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring Massive Multitask Language Understanding.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Hua, Wenxiao Wang, Zihui Xue, Sucheng Ren, Yue Wang, and Hang Zhao. 2021. On Feature Decorrelation in Self-Supervised Learning. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9578–9588, Montreal, QC, Canada. IEEE.

Hanxun Huang, Ricardo J. G. B. Campello, Sarah Monazam Erfani, Xingjun Ma, Michael E. Houle, and James Bailey. 2023. LDReg: Local Dimensionality Regularized Self-Supervised Learning.

Katharina Hämmerl, Alina Fastowski, Jindřich Libovický, and Alexander Fraser. 2023. Exploring Anisotropy and Outliers in Multilingual Language Models for Cross-Lingual Semantic Sentence Similarity. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7023–7037, Toronto, Canada. Association for Computational Linguistics.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised Dense Information Retrieval with Contrastive Learning. *Transactions on Machine Learning Research*.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. 2021. Understanding Dimensional Collapse in Contrastive Self-supervised Learning.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

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.

Olga Kovaleva, Saurabh Kulshreshtha, Anna Rogers, and Anna Rumshisky. 2021. BERT Busters: Outlier Dimensions that Disrupt Transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3392–3405, Online. Association for Computational Linguistics.

Artur Kulmizev, Vinit Ravishankar, Mostafa Abdou, and Joakim Nivre. 2020. Do neural language models show preferences for syntactic formalisms? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4077–4091, 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 Ali Farhadi. 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*, 7:453–466.

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.

Siran Li, Linus Stenzel, Carsten Eickhoff, and Seyed Ali Bahrainian. 2025. Enhancing retrieval-augmented generation: A study of best practices. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6705–6717, Abu Dhabi, UAE. Association for Computational Linguistics.

Peiyang Liu, Xi Wang, Sen Wang, Wei Ye, Xiangyu Xi, and Shikun Zhang. 2021. Improving embedding-based large-scale retrieval via label enhancement. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 133–142, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are Sixteen Heads Really Better than One? In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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. 2022. Sentence-T5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the association for computational linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. 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.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation Learning with Contrastive Predictive Coding. *arXiv preprint*. ArXiv:1807.03748 [cs].

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830.

Giovanni Puccetti, Anna Rogers, Aleksandr Drozd, and Felice Dell'Orletta. 2022. Outlier Dimensions that Disrupt Transformers are Driven by Frequency. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1286–1304, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2025. Qwen2.5 Technical Report. *arXiv preprint*. ArXiv:2412.15115 [cs].

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(140):1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Vikas Raunak, Vivek Gupta, and Florian Metze. 2019. Effective Dimensionality Reduction for Word Embeddings. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 235–243, Florence, Italy. Association for Computational Linguistics.

Anton Razzhigaev, Matvey Mikhalchuk, Elizaveta Goncharova, Ivan Oseledets, Denis Dimitrov, and Andrey Kuznetsov. 2024. The Shape of Learning: Anisotropy and Intrinsic Dimensions in Transformer-Based Models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 868–874, St. Julian's, Malta. Association for Computational Linguistics.

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.

William Rudman, Catherine Chen, and Carsten Eickhoff. 2023. Outlier Dimensions Encode Task Specific Knowledge. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14596–14605, Singapore. Association for Computational Linguistics.

William Rudman and Carsten Eickhoff. 2023. Stable Anisotropic Regularization.

William Rudman, Nate Gillman, Taylor Rayne, and Carsten Eickhoff. 2022. IsoScore: Measuring the Uniformity of Embedding Space Utilization. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3325–3339, Dublin, Ireland. 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.

Sam Sattarzadeh, Mahesh Sudhakar, Anthony Lem, Shervin Mehryar, Konstantinos N. Plataniotis, Jongseong Jang, Hyunwoo Kim, Yeonjeong Jeong, Sangmin Lee, and Kyunghoon Bae. 2021. Explaining Convolutional Neural Networks through Attribution-Based Input Sampling and Block-Wise Feature Aggregation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):11639–11647. Number: 13.

Sofia Serrano and Noah A. Smith. 2019. Is Attention Interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3319–3328. PMLR. ISSN: 2640-3498.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models.

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.

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).

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. Stroudsburg, PA, USA. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *arXiv preprint*. ArXiv:2212.03533 [cs].

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.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th international conference on machine learning*, ICML'20. JMLR.org. Number of pages: 11 tex.articleno: 921.

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.

Siyue Wu, Hongzhan Chen, Xiaojun Quan, Qifan Wang, and Rui Wang. 2023. AD-KD: Attribution-Driven Knowledge Distillation for Language Model Compression. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8449–8465, Toronto, Canada. Association for Computational Linguistics.

Chenghao Xiao, Yang Long, and Noura Al Moubayed. 2023. On isotropy, contextualization and learning dynamics of contrastive-based sentence representation learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12266–12283, Toronto, Canada. 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.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Gaifan Zhang, Yi Zhou, and Danushka Bollegala. 2024a. Evaluating Unsupervised Dimensionality Reduction Methods for Pretrained Sentence Embeddings. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6530–6543, Torino, Italia. ELRA and ICCL.

Ruochen Zhang, Qinan Yu, Matianyu Zang, Carsten Eickhoff, and Ellie Pavlick. 2024b. The Same but Different: Structural Similarities and Differences in Multilingual Language Modeling.

Rui Zheng, Bao Rong, Yuhao Zhou, Di Liang, Sirui Wang, Wei Wu, Tao Gui, Qi Zhang, and Xuanjing Huang. 2022. Robust Lottery Tickets for Pre-trained Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2211–2224, Dublin, Ireland. Association for Computational Linguistics.

A Appendix

A.1 Relative Performances by Different Seeds

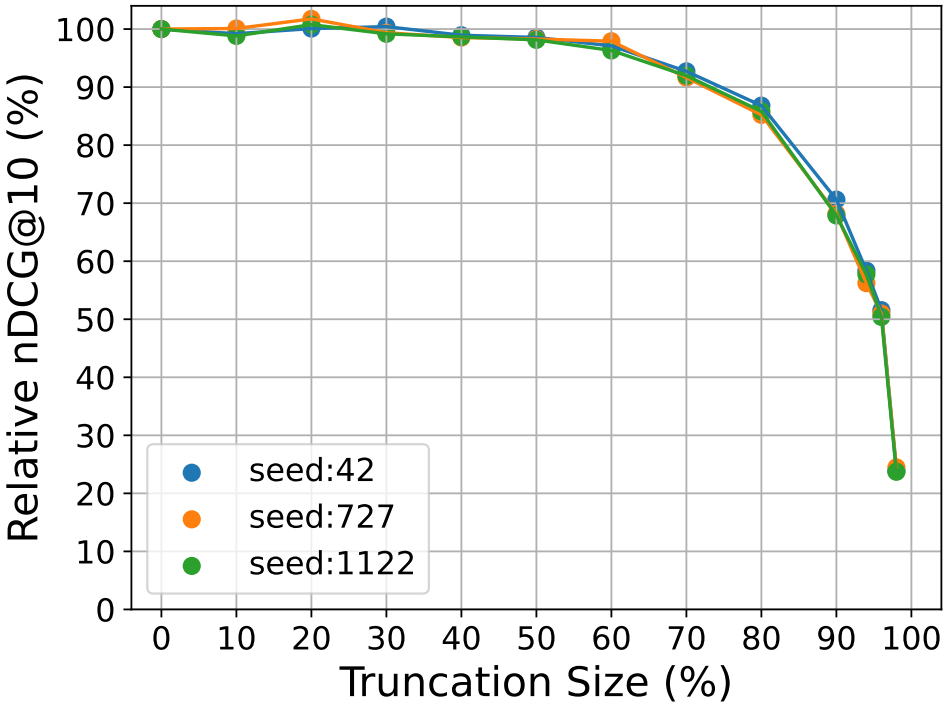


Figure 7: Relative performance of three differently-seeded models on NanoBEIR benchmark with different truncation sizes.

Here, we investigate whether the embeddings produced by all of the differently-seeded models during contrastive training achieve high relative performance by fine-tuning the T5's encoder with a contrastive learning objective as done in §3.1 with three different random seeds. The result is shown in Fig. 7. All three model instances have almost identical shifts in relative performance with different degrees of truncations. This result allows us to conclude that our observation, the embeddings' high robustness to the truncation operation, is present regardless of the randomness during contrastive training.

A.2 LLMs and Truncated Representations

Model	Method	COPA F1 (Relative)	DROP F1 (Relative)	HellaSwag Acc (Relative)
Llama	Full	0.194 (1.000)	0.194 (1.000)	0.681 (1.000)
	First	0.094 (0.487)	0.094 (0.487)	0.580 (**0.852**)
	Last	0.038 (0.198)	0.038 (0.198)	0.586 (**0.861**)
Qwen	Full	0.003 (1.000)	0.003 (1.000)	0.718 (1.000)
	First	0.001 (0.375)	0.001 (0.375)	0.709 (**0.988**)
	Last	0.001 (0.458)	0.001 (0.458)	0.709 (**0.987**)

Table 4: Performance on three benchmark datasets when the last hidden representations and the unembedding layer are reduced by half. Relative performance (scores in parenthesises) is bolded when it reaches 80% of the original performance.

Table 4 is a complementary table to Table 3, showing the impact of representation truncation on the three remaining datasets from §2.

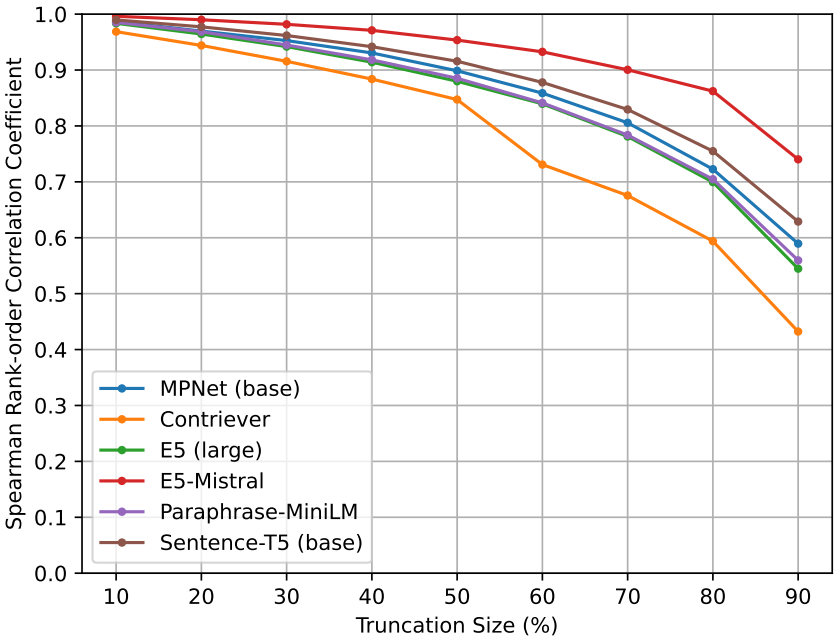


Figure 8: Spearman's rank-order correlation coefficient between the two rankings of documents for each query in NanoBEIR produced by full-sized embeddings and truncated embeddings.

A.3 Document Ranking by Truncated Embeddings

We compare two sets of document rankings. The first set is a result of standard text embedding-based retrieval, where we use the original full-sized embeddings to compute similarity between queries and documents. The second set is produced by using embeddings of which the last K% of the dimensions are truncated. We compute Spearman's rank-order correlation between the two and see how similar the rankings produced by truncated embeddings are to the original embeddings' rankings. The resulting curve is shown in Fig. 8. Similarly to our findings in §2, the behaviour of truncated embeddings is close to the original embeddings.

A.4 Truncated Embeddings' Performance Per Dataset

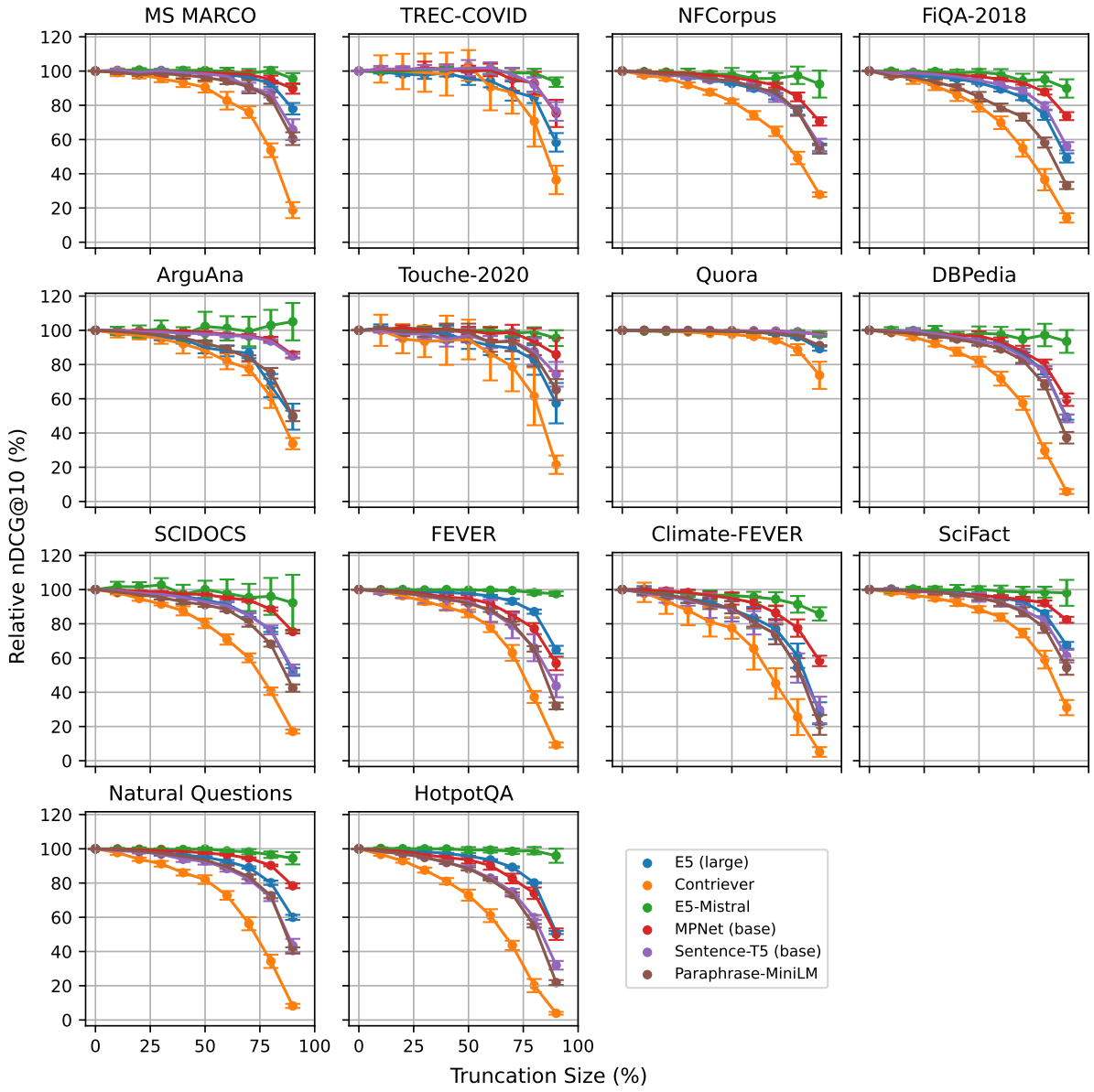


Figure 9: The relative performance achieved by randomly truncated embeddings per dataset from BEIR and NanoBEIR for E5-Mistral.

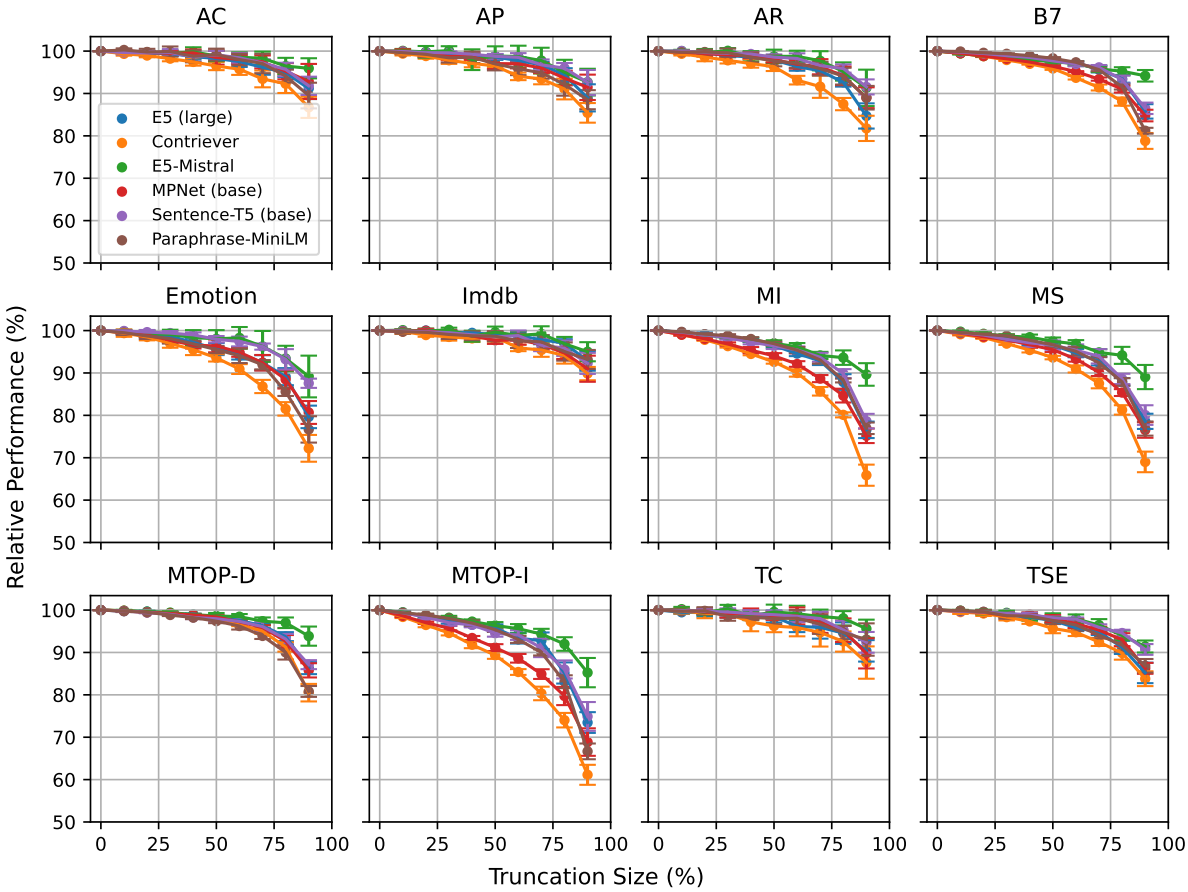


Figure 10: The relative performance achieved by randomly truncated embeddings per dataset from MTEB.

A.5 Effective Use of Representation Space (Figures for BERT)

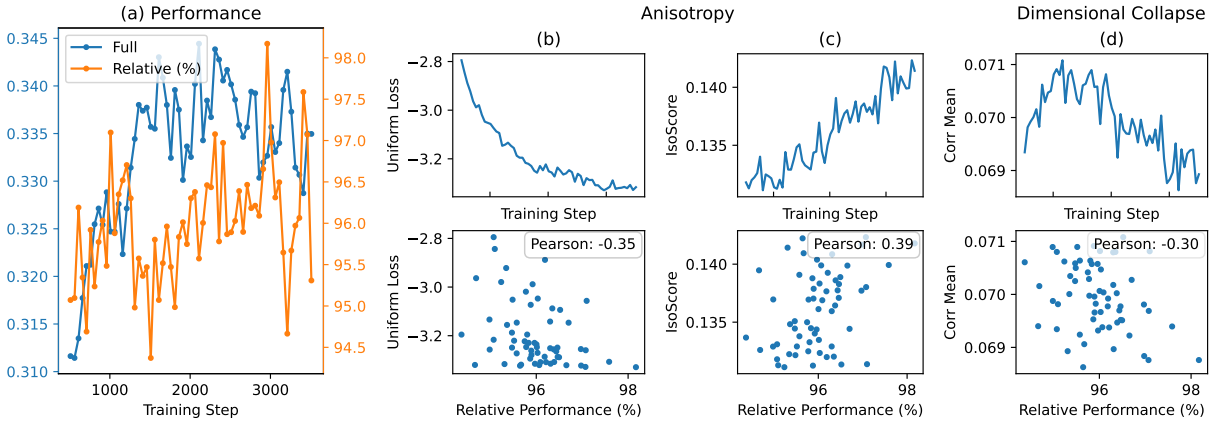


Figure 11: As a result of contrastive learning for BERT, downstream task performance increases (a: Full), and the use of embedding space measured through Uniform Loss (↓) and IsoScore (↑) for anisotropy (b, c: top) and Corr Mean (↓) for dimensional collapse (d: top) also improves. However, the relative performance does not change over the training (a: Relative), therefore, there is no strong correlation between relative performance and representation quality measures (b, c, d: bottom).

A.6 Effect of Removing Only Degrading (or Improving) Dimensions

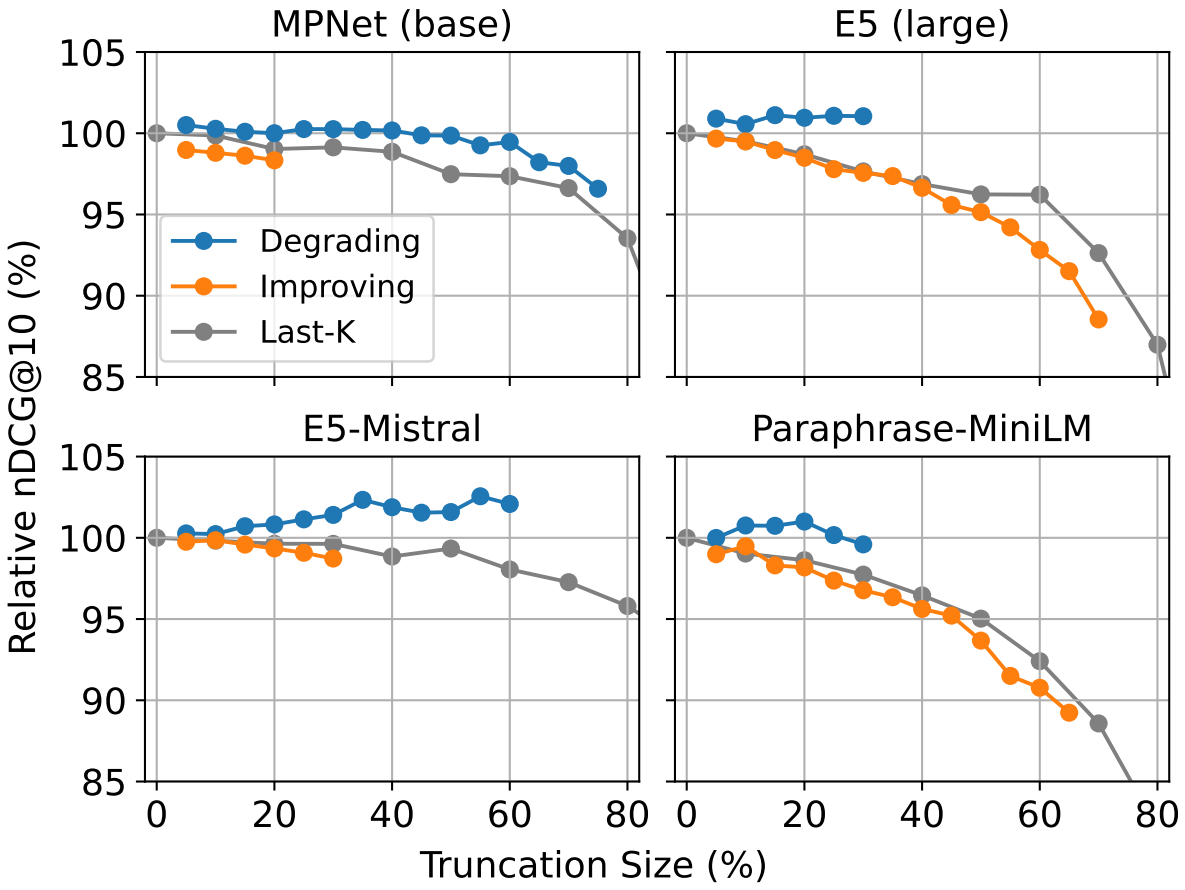


Figure 12: When only the degrading dimensions are removed (blue plot), the performance improves over the original embeddings first, and as more are removed, the performance starts to decay, however, more slowly than the last-k truncation. On the other hand, when only the improving dimensions are removed (orange plot), the performance decreases rapidly.

A.7 E5-Mistral's Outlier Dimensions.

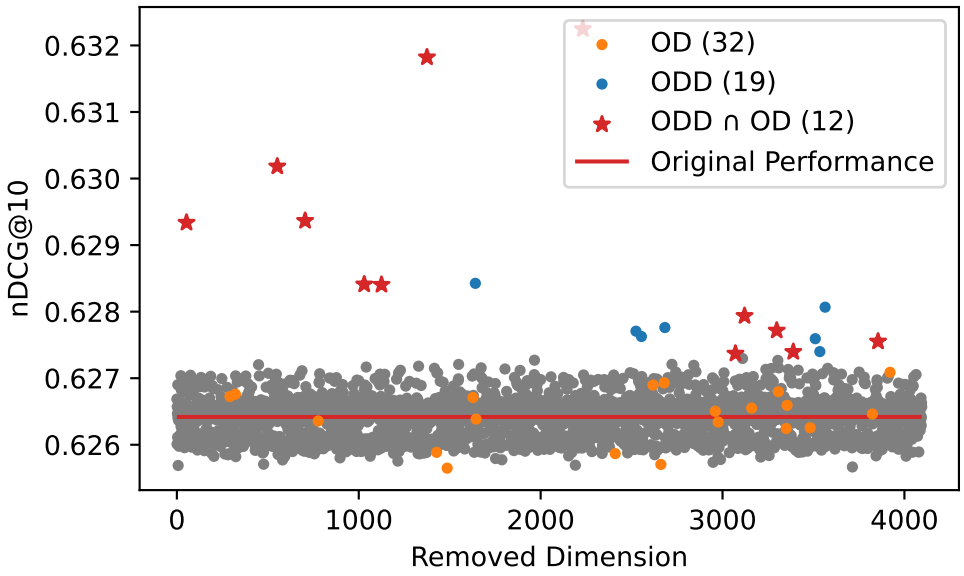


Figure 13: E5-Mistral's retrieval performance without one dimension. Outlier dimensions (ODs) take abnormal values within embeddings as defined in §3.3. Outlier degrading dimensions (ODDs) have negative impacts that are at least 3σ from the mean of performance. The numbers in parentheses indicate the count of each type of dimension.

A.8 Resources Used for Our Experiments

Name	Params	Emb Size	Licence
MPNet (base)	109M	768	Apache license 2.0
Contriever (Izacard et al., 2022)	110M	768	Attribution-NonCommercial 4.0 International
E5 (large) (Wang et al., 2022)	335M	1024	MIT
E5-Mistral (Wang et al., 2024)	7B	4096	MIT
Paraphrase-MiniLM (Reimers and Gurevych, 2019)	17M	384	Apache license 2.0
Sentence-T5 (base) (Ni et al., 2022)	110M	768	Apache license 2.0

Table 5: List of models used in our study.

	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
	ToxicConversationsClassification (cjadams et al., 2019)	Social, Written	CC-by-4.0
	TweetSentimentExtractionClassification (Maggie et al., 2020)	Social, Written	N/A

Table 6: A list of datasets used in our evaluation.