Redundancy, Isotropy, and Intrinsic Dimensionality of Prompt-based Text Embeddings

Hayato Tsukagoshi **Ryohei Sasano**
Graduate School of Informatics, Nagoya University
tsukagoshi.hayato.r2@s.mail.nagoya-u.ac.jp sasano@i.nagoya-u.ac.jp

Abstract

Prompt-based text embedding models, which generate task-specific embeddings upon receiving tailored prompts, have recently demonstrated remarkable performance. However, their resulting embeddings often have thousands of dimensions, leading to high storage costs and increased computational costs of embedding-based operations. In this paper, we investigate how post-hoc dimensionality reduction applied to the embeddings affects the performance of various tasks that leverage these embeddings, specifically classification, clustering, retrieval, and semantic textual similarity (STS) tasks. Our experiments show that even a naive dimensionality reduction, which keeps only the first 25% of the dimensions of the embeddings, results in a very slight performance degradation, indicating that these embeddings are highly redundant. Notably, for classification and clustering, even when embeddings are reduced to less than 0.5% of the original dimensionality the performance degradation is very small. To quantitatively analyze this redundancy, we perform an analysis based on the intrinsic dimensionality and isotropy of the embeddings. Our analysis reveals that embeddings for classification and clustering, which are considered to have very high dimensional redundancy, exhibit lower intrinsic dimensionality and less isotropy compared with those for retrieval and STS.

1 Introduction

Text embeddings are a foundational component of many natural language processing (NLP) applications, including document retrieval, retrieval-augmented generation (RAG), and text clustering. Recent advances in large language models (LLMs) renewed interest in text representation learning, owing to their strong language understanding and generalization capabilities (Muennighoff, 2022; Ni et al., 2022b,a; Yano et al., 2024; Springer et al., 2024; Jiang et al., 2024). Among these, prompt-based text embedding models, which produce task-specific embeddings by incorporating natural language instructions or task descriptions, have demonstrated remarkable performance on various tasks (Su et al., 2023; Asai et al., 2023; Lee et al., 2024a,b; Wang et al., 2022; Li et al., 2023; Xiao et al., 2024). However, prompt-based models typically generate embeddings with thousands of dimensions, leading to high storage costs and increased computational costs of embedding-based operations. For instance, E5-mistral (Wang et al., 2024a), a model obtained by fine-tuning Mistral-7B (Jiang et al., 2023), produces 4096-dimensional embeddings. Reducing the dimensionality of LLM-based text embeddings through post-processing could thus offer substantial practical benefits.

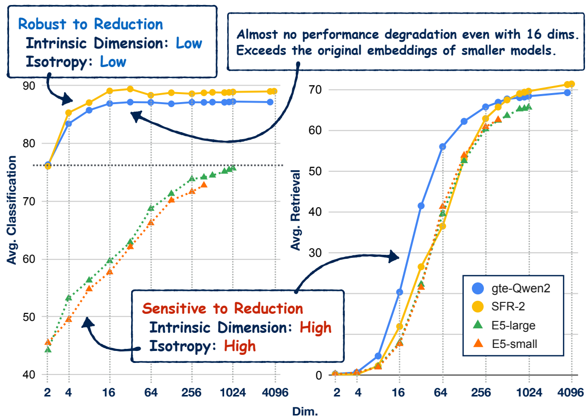

Figure 1: The performance on classification and retrieval tasks under dimensionality reduction. The models used in the experiments are described in Section 2.

In this work, we show that prompt-based text embedding models can maintain surprisingly strong performance even when their dimensionality is substantially reduced. Figure 1 summarizes our findings. Even a naive dimensionality reduction that keeps only the first 25% of the embedding dimensions results in almost no performance degradation

Findings of the Association for Computational Linguistics: ACL 2025, pages 25915–25930
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

across a range of tasks, indicating that these embeddings are highly redundant. Notably, for classification task, reducing embeddings to less than 0.5% of the original dimensionality can almost preserve their original performance. Furthermore, we find that the embeddings after dimensionality reduction perform better than the same-dimensional embeddings produced by smaller models. We also find that the robustness of these models to dimensionality reduction varies by task type; while classification and clustering exhibit a more gradual performance decline, with the extent of degradation varying across models, retrieval and STS tasks tend to experience a more rapid drop in performance, with several models showing similar trends.

To investigate why such a significant reduction is feasible, we quantitatively assess the redundancy in the generated embeddings. Specifically, we analyze the intrinsic dimensionality (ID) and isotropy of representations derived from different task-specific prompts using IsoScore (Rudman et al., 2022). Our findings reveal that prompt-based text embedding models produce distinct representation properties depending on the task prompt. For classification and clustering tasks, we observe lower intrinsic dimensionality and less isotropic distributions, which correlates with their high redundancy and robustness to even drastic dimensionality reduction. In contrast, for retrieval and STS tasks—where fine-grained similarity is critical—embeddings exhibit higher intrinsic dimensionality and more isotropic distributions in the embedding space. Moreover, we find a relationship between measures such as ID and IsoScore and the robustness of embeddings to dimensionality reduction; embeddings for classification and clustering tasks are highly redundant and remain robust even under drastic reduction, whereas those for retrieval and STS tasks are less redundant and degrade more significantly when dimensions are reduced.

2 Robustness of Text Embeddings for Dimensionality Reduction

In this section, we demonstrate that prompt-based text embeddings exhibit high robustness to dimensionality reduction in specific tasks and reveal that their embeddings contain redundancy.

2.1 Evaluation Tasks

To conduct a comprehensive analysis across various tasks, we evaluate text embedding models using the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2023). In this study, we use several English datasets from four categories: classification, clustering, retrieval, Semantic Textual Similarity (STS).

Classification Classification tasks evaluate the quality of text embeddings by training a logistic regression classifier to predict the labels of given texts based on their corresponding embeddings. The logistic regression classifier is trained on the training set and evaluated on the test set. We use the default settings to train the logistic regression classifier without modifications. Since the evaluation metrics vary by task, we adopt the default metrics. In this study, we employ five tasks: AmazonCounterfactualClassification (O'Neill et al., 2021), AmazonPolarityClassification (McAuley and Leskovec, 2013), AmazonReviewsClassification (Keung et al., 2020), ImdbClassification (Maas et al., 2011), and ToxicConversationsClassification.[1]

Clustering Clustering tasks evaluate how well the clusters formed based on distances in the embedding space align with the ground-truth clusters. For evaluation, we use the V-Measure metric (Rosenberg and Hirschberg, 2007), which is the default evaluation metric in MTEB. In this study, we use three tasks; RedditClustering, StackExchangeClustering (Geigle et al., 2021), and ArxivClusteringS2S.[2]

Retrieval[3] Retrieval tasks assess document retrieval performance based on embeddings. For evaluation, search queries and a collection of documents are encoded into embeddings. The similarity between query and document embeddings is computed, and retrieval performance is assessed by checking whether the relevant document appears among the top-ranked results. Cosine similarity is a commonly used metric, and we use it in this study as well. We use nDCG@10 as

[1] https://kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification

[2] https://www.kaggle.com/datasets/Cornell-University/arxiv

[3] For retrieval tasks, encoding the entire document collection, which comprises millions of examples, into embeddings for each experiment is computationally infeasible. Therefore, we use the down-sampled version officially provided by MTEB for these evaluations. This version includes only the document sets corresponding to 250 hard negatives collected for each search query using BM25 or mE5 (Wang et al., 2024b), and the maximum number of examples per dataset has been reduced to 1,000 (see https://github.com/embeddings-benchmark/mteb/pull/1236).

the evaluation metric. For the evaluation datasets, we use MIRACL (Zhang et al., 2022), Quora,[4] HotpotQA (Yang et al., 2018), DBPedia (Hasibi et al., 2017), Natural Questions (Kwiatkowski et al., 2019), and MS MARCO (Nguyen et al., 2016).

STS Semantic textual similarity (STS) tasks evaluate how well the semantic similarity between sentence pairs, as determined by their embeddings, correlates with human-annotated similarity scores. For evaluation, we adopt Spearman's rank correlation coefficient, in line with previous studies (Reimers and Gurevych, 2019; Tsukagoshi et al., 2021; Gao et al., 2021). In this study, we utilize seven tasks: STS12–16 (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS Benchmark (Cer et al., 2017), and SICK-R (Marelli et al., 2014).

2.2 Experimental Models

Prompt-based text embedding models can be categorized into two types; instruction-based text embedding models, which use natural language instructions as prompts (Su et al., 2023; Asai et al., 2023; Wang et al., 2024a; Lee et al., 2024a), and prefix-based text embedding models, which add pre-defined task-specific prefixes to the beginning of texts (Wang et al., 2022, 2024b; Nussbaum et al., 2024; Li et al., 2023; Xiao et al., 2024).

In general, instruction-based text embedding models leverage the in-context learning capabilities of large language models (LLMs) and are often built by fine-tuning LLMs. In contrast, prefix-based text embedding models are typically constructed by fine-tuning smaller models, such as BERT (Devlin et al., 2019), using large-scale contrastive learning. We include both types of models in our experiments. For example, the instruction-based models consist of gte-Qwen2 with 7.6B parameters and an embedding dimension of 3,584, E5-mistral (Wang et al., 2024a) and SFR-Embedding-2_R[5] each with 7.1B parameters and an embedding dimension of 4,096, and mE5-large-inst (Wang et al., 2024b) with 560M parameters and an embedding dimension of 1,024. These models incorporate task-specific instructions to generate embeddings. Other models used in our experiments include Unsup-SimCSE (Gao et al., 2021), the small and large variants of E5 (Wang et al., 2022), and Nomic (Nussbaum et al., 2024).

[4] `https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs`

[5] `https://huggingface.co/Salesforce/SFR-Embedding-2_R`

Unsup-SimCSE is a fine-tuned BERT-large with contrastive learning, while E5 uses two prefixes, "`query:`" and "`passage:`." Nomic adapts to different tasks by employing different prefixes. Specifically, the prefix "`search_query:`" is used for retrieval queries, "`search_document:`" for retrieval documents, "`classification:`" for classification tasks, "`clustering:` for clustering tasks, and for tasks such as STS in which the semantic content of the text is embedded, no prefix is used. The task-specific prompts used for the instruction-based text embedding models are listed in Appendix A, and more detailed descriptions of each model are provided in Appendix B.

2.3 Evaluation Method

For each text embedding model, we iteratively reduce the dimensionality of the embeddings and evaluate the performance to observe the relationship between dimensionality reduction and performance degradation. While several methods for dimensionality reduction, such as principal component analysis, are conceivable, this study simply reduces the dimensionality by taking the first $d \in \mathbb{Z}_{>0}$ dimensions of the output embeddings. We do not normalize the output embeddings.

It is worth noting that, methods like matryoshka representation learning (Kusupati et al., 2022) exist to enable dimensionality reduction by simply taking the first d dimensions of embeddings. To ensure that the results obtained in this study are not attributable to such specific methods, we compared the performance when reducing dimensionality by randomly taking d dimensions rather than taking the first d dimensions. The results showed no differences that would affect the observed results. We further conducted experiments using more sophisticated dimensionality reduction methods such as PCA; however, these did not reveal significant differences in the general trends. Experimental results for dimensionality reduction methods other than taking the first d dimensions, including taking the random d dimensions, PCA (Abdi and Williams, 2010), UMAP (McInnes et al., 2020), and Isomap (Tenenbaum et al., 2000) are presented in Appendix C.

2.4 Experimental Results

Regarding the performance trends associated with dimensionality reduction, the results for the various tasks are presented in Figure 2 for classification tasks, Figure 3 for clustering tasks, Figure 4 for

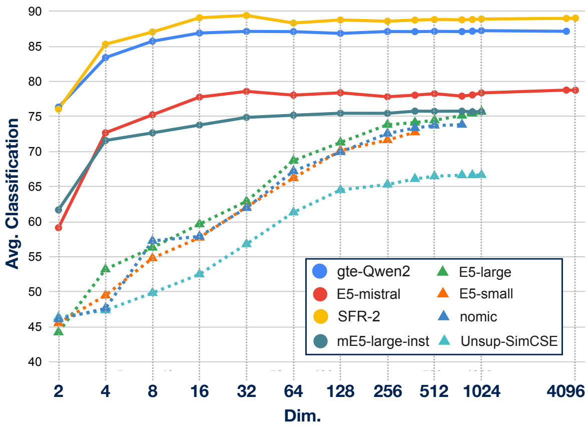

Figure 2: The relationship between the number of dimensions and the average performance on classification tasks. The horizontal axis is logarithmic. Circular markers with solid lines correspond to instruction-based text embedding models, whereas triangular markers with dashed lines correspond to other models.

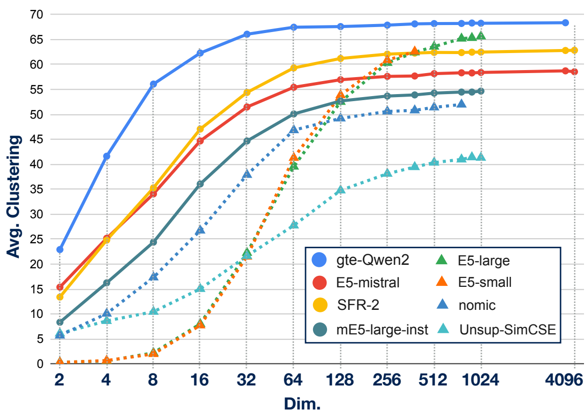

Figure 3: The relationship between the number of dimensions and the average performance on clustering tasks. Other details are the same as in Figure 2.

retrieval tasks, and Figure 5 for STS tasks.

Classification For classification tasks, we observed that the performance trends differ between instruction-based models and other models. For instruction-based models, the degradation in performance was remarkably gradual. In particular, both gte-Qwen2 and SFR-2 exhibited minimal performance decline when the embedding dimensionality was reduced to merely 8 dimensions (0.2% of the original dimensions). Notably, gte-Qwen2 achieved a score of 76.34 with just 2 dimensions, surpassing the 75.69 score obtained using the full 1024-dimensional embeddings produced by E5-large. In contrast, models like E5-large exhibited a monotonic decrease in performance as the dimensionality was reduced. These results suggest

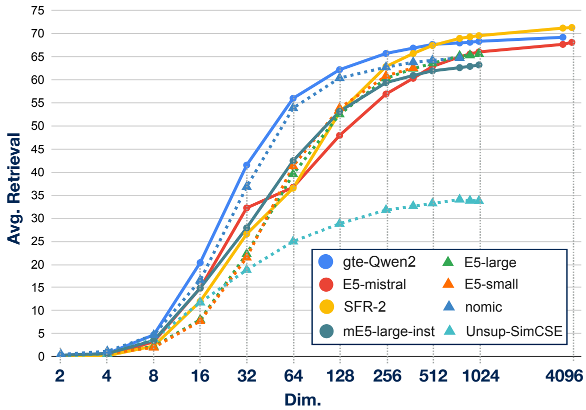

Figure 4: The relationship between the number of dimensions and the average performance on retrieval tasks. Other details are the same as in Figure 2.

that instruction-based text embeddings not only generate high-quality representations for text classification, but also that the minimal performance degradation observed after dimensionality reduction implies that these embeddings exhibit significant redundancy.

Clustering The trends in clustering tasks differ slightly from those observed in classification tasks. Although performance degradation is relatively noticeable in clustering tasks, instruction-based models remain robust to dimensionality reduction. Specifically, we observed that LLM-based text embedding models exhibit negligible degradation even when the dimensionality is reduced to around 128 dimensions (less than 4% of the original dimensionality). On the other hand, in contrast to the trends observed in classification tasks, while E5 achieves high performance when using the full-dimensional embeddings, for both E5-large and E5-small the performance degradation becomes substantial. When gte-Qwen2 embeddings are reduced to 128 dimensions (3.6% of the original), the performance degradation is only about 0.8 points. In contrast, when E5-large embeddings are reduced to 128 dimensions (12.5% of the original), the performance drops by approximately 13 points, suggesting that the E5 embeddings contain little redundancy.

Retrieval and STS We observed that the performance degradation trends in retrieval and STS tasks were largely similar across all models, with performance consistently declining as the dimensionality was reduced, in contrast to the trends observed in classification and clustering tasks. That said, when the dimensionality of gte-Qwen2 embeddings was

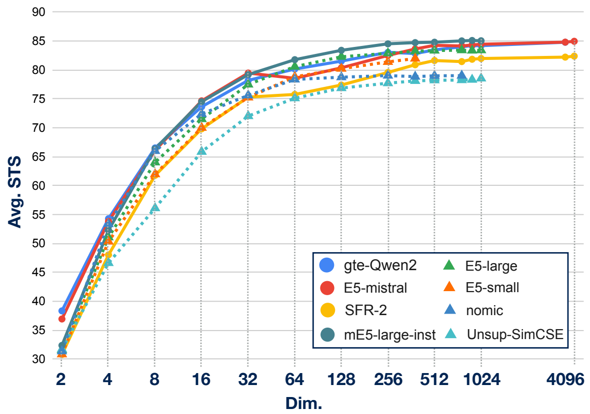

Figure 5: The relationship between the number of dimensions and the average performance on STS tasks. Other details are the same as in Figure 2.

reduced to 512 dimensions (approximately 14% of the original), the performance loss in retrieval tasks was only about 1.5 points, remaining relatively limited.

Overall After aligning the embedding dimensionality, those produced by larger, higher-performing models consistently outperformed embeddings from smaller models. Moreover, for classification and clustering tasks, we found that instruction-based text embeddings can maintain high performance even when only the first several dimensions of their embeddings were used, suggesting that they may be redundant. Across tasks, classification was the most resilient to dimensionality reduction, followed by clustering, whereas retrieval and STS tasks were more sensitive. Altogether, these results indicate that the extent of embedding redundancy varies by task.

3 Intrinsic Dimensionality and Isotropy of Prompt-based Text Embeddings

To investigate why the robustness to dimensionality reduction varies across tasks, we quantitatively evaluate the redundancy of the embeddings.

3.1 Evaluation Method

We assess the degree of redundancy in the generated text embeddings as tasks vary. Specifically, we measure the intrinsic dimension (ID) and isotropy as indicators of redundancy, and we analyze how these metrics change as the prompt is varied across a collection of texts.

Intrinsic Dimension The intrinsic dimension refers to the number of dimensions required to capture the essential structure of data representations. Several methods have been proposed for estimating the intrinsic dimension (Bruske and Sommer, 1998; Fukunaga and Olsen, 1971; Levina and Bickel, 2004); among these, we employ TwoNN (Facco et al., 2017). TwoNN estimates the intrinsic dimension by analyzing the ratio of distances between each point and its two nearest neighbors changes with a set of embeddings. In high-dimensional spaces, the ratio of the distances to the first and second nearest neighbors follows a Pareto distribution for points uniformly distributed on a d-dimensional manifold (Ansuini et al., 2019). TwoNN uses this property to estimate the intrinsic dimension. Notably, TwoNN is robust even when the underlying manifold is curved or the sampling density is nonuniform, and it is computationally efficient. We use the Python library `scikit-dimension`[6] to compute the intrinsic dimensions via TwoNN.

IsoScore IsoScore (Rudman et al., 2022) is a metric used to evaluate the isotropy of embeddings. Isotropy refers to the extent to which embeddings are uniformly distributed across the entire embedding space without bias toward specific dimensions. Intuitively, IsoScore is computed by calculating the variance-covariance matrix of the embedding representations, normalizing it, and then measuring its deviation from the identity matrix. IsoScore ranges from 0 to 1, with values close to 1 indicating that the embeddings are distributed isotropically and values near 0 indicating anisotropic distribution.[7]

Evaluation Procedure We randomly sampled 10,000 texts from English Wikipedia and obtained embeddings for each model and prompt.[8] The models, instructions, and prefixes used are essentially the same as those described in Section 2. Additionally, we included the BERT-large `[CLS]` embedding and the average of output contextualized

[6] `https://github.com/scikit-learn-contrib/scikit-dimension`

[7] Although IsoScore* (Rudman and Eickhoff, 2024) was introduced to stabilize IsoScore computations on small datasets and enable full differentiability, our study does not involve training new embedding models; rather, we focus on evaluating the isotropy of embeddings produced by existing models. Since IsoScore reliably computes stable scores when the number of data samples exceeds the embedding dimensionality—and because it remains computationally efficient without requiring additional regularization—we employed the original IsoScore in our experiments.

[8] The texts from English Wikipedia were extracted from the <p> tags in the HTML dump at `https://dumps.wikimedia.org/other/enterprise_html/`. Some texts contain multiple sentences, while others may be shorter than a full sentence.

Prompt Type	gte-Qwen2		E5-mistral		SFR-2		mE5-large-inst		nomic	
	ID	IsoScore	ID	IsoScore	ID	IsoScore	ID	IsoScore	ID	IsoScore
Classification	22.02	.0052	22.26	.0057	37.03	.0077	21.85	.0191	27.75	.1556
Clustering	10.78	.0058	13.01	.0060	16.29	.0138	17.29	.0405	26.25	.1362
Retrieval										
Query	31.90	.0779	51.36	.0761	81.38	.1117	36.59	.1750	34.74	.2112
Passage	35.94	.0813	36.69	.0332	35.07	.0555	35.58	.0752	33.78	.1930
STS	38.47	.0784	34.07	.0439	41.69	.0533	34.96	.1400	32.84	.2127

Table 1: Intrinsic dimensions and IsoScore for models using task-specific prompts, by model and prompt type.

Model	Prompt	ID	IsoScore
E5-small	query:	41.57	.4419
	passage:	37.60	.3905
E5-large	query:	42.44	.2022
	passage:	38.50	.1977
Unsup-SimCSE		27.01	.1611
BERT (CLS)		20.78	.0186
BERT (Mean)		17.56	.0973

Table 2: Intrinsic dimensions and IsoScore for models without task-specific prompts, by model and prompt.

word embeddings in our experiments. It is worth noting that, for instruction-based text embedding models, different prompts are used for each task even within the same task type. Therefore, we compute intrinsic dimensions and IsoScore for each prompt and then take the average for each task type. In retrieval tasks, different instructions or prefixes may be used for queries and documents. Hence, we calculate the intrinsic dimension and IsoScore separately for each. As a result, the prompt types consist of retrieval queries, retrieval documents, STS, classification, and clustering.

3.2 Experimental Results

The results for the instruction-based text embeddings are shown in Table 1, and the results for the other text embeddings are shown in Table 2. For all models, the intrinsic dimensions were significantly smaller than the actual dimensions of the embeddings. Larger models exhibited lower IsoScore values, whereas smaller models demonstrated relatively high isotropy.

Focusing on Table 1, instruction-based text embedding models tended to have smaller intrinsic dimensions and lower IsoScore values when prompts for classification or clustering tasks were used. In contrast, prompts for retrieval queries, retrieval documents, or STS tasks resulted in higher intrinsic dimensions and IsoScore values. When comparing instruction-based models, those built on LLMs exhibited greater differences in intrinsic dimensions and IsoScore values between classification/clustering tasks and retrieval/STS tasks. Furthermore, instruction-based text embedding models (e.g., gte-Qwen2, E5-mistral, SFR-2, mE5-large-inst) showed an average difference of more than 10 in intrinsic dimension and approximately a tenfold difference in IsoScore between embeddings generated for retrieval or STS tasks and those for classification or clustering tasks. That is, embeddings for classification and clustering tasks are relatively anisotropic, whereas those for retrieval and STS tasks are comparatively isotropic, indicating that embeddings for classification and clustering tasks are relatively more redundant.

Table 2 illustrates that both text embedding models not based on prompts and prefix-based models such as E5 generally exhibited relatively high intrinsic dimension and IsoScore values. Both Unsup-SimCSE and E5-large showed higher intrinsic dimensions and IsoScore values than the original BERT-large, which aligned with previous research indicating that contrastive learning enhanced the uniformity of embeddings (Gao et al., 2021). E5 consistently demonstrated high values regardless of the prefix, often exhibiting larger intrinsic dimensions than those observed in LLM-based text embeddings. These findings suggest that E5, which employed the prefix for diverse tasks such as retrieval queries and text classification, might generate embeddings with lower redundancy in order to preserve a broader range of information. This distinction aligns with the differing requirements of downstream tasks. Retrieval tasks require capturing subtle semantic relationships between sentences or documents, necessitating the retention of a substantial amount of information within the embeddings. In contrast, classification and clustering tasks require only the details relevant to specific classes. Indeed, our observations indicate that prompt-based embedding models adapt to these task characteristics by producing embeddings with higher redundancy for classification and cluster-

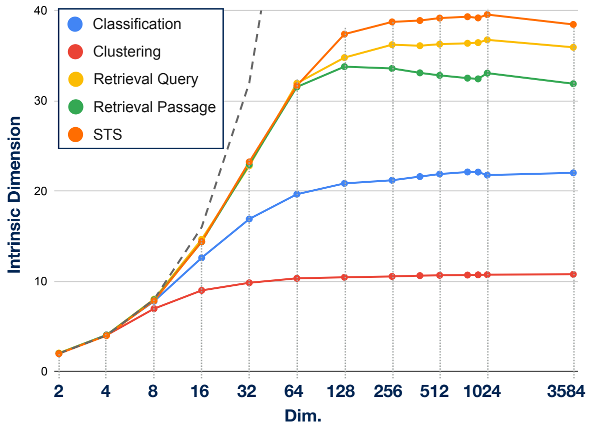

Figure 6: ID under dimensionality reduction. The dashed line represents the actual dimensions.

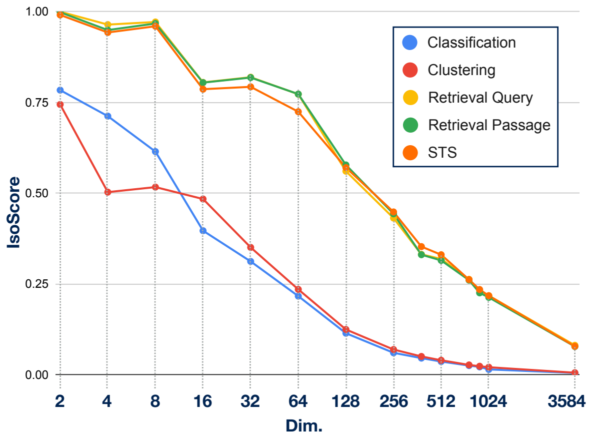

Figure 7: IsoScore under dimensionality reduction.

ing tasks, while yielding embeddings with lower redundancy for retrieval and STS tasks.

ID and Isotropy with Dimensionality Reduction As in Section 2, we performed dimensionality reduction on the embeddings and evaluated the changes in intrinsic dimension and isotropy. We measured the ID and IsoScore at each dimension using embeddings from gte-Qwen2 for each prompt type, and the results are shown in Figure 6 and Figure 7. Regarding ID, the ordering of the IDs for the full-dimensional embeddings did not change with dimensionality reduction; across all prompt types, the IDs remained nearly stable until approximately 128 dimensions. Regarding IsoScore, the trends in IsoScore differed between embeddings for classification/clustering tasks and those for retrieval/STS tasks. Specifically, while the IsoScore for embeddings intended for classification and clustering remained around 0.75 even when reduced to 2 dimensions, the IsoScore for embeddings intended for retrieval and STS tasks nearly reached 1, indicating that the corresponding subspaces were isotropic.

4 Related Work

4.1 Text Embeddings

Early research on text embeddings focused on deriving sentence embeddings from word embeddings (Shen et al., 2018; Mu and Viswanath, 2018; Arora et al., 2017; Ethayarajh, 2018), while later methods such as InferSent (Conneau et al., 2017), Sentence-BERT (Reimers and Gurevych, 2019), and Supervised SimCSE (Gao et al., 2021) fine-tuned pre-trained language models on NLI datasets (Bowman et al., 2015; Williams et al., 2018) for richer semantic representations. Moreover, to capture various types of information beyond semantics, prompt-based text embedding models tailor embeddings to specific tasks using prefixes (Wang et al., 2022, 2024b; Nussbaum et al., 2024; Li et al., 2023; Xiao et al., 2024) or natural language instructions (Su et al., 2023; Asai et al., 2023; Wang et al., 2024a; Lee et al., 2024a; Muennighoff et al., 2024).

Prefix-based approaches like E5 (Wang et al., 2022), multilingual E5 (Wang et al., 2024b), GTE (Li et al., 2023), and BGE (Xiao et al., 2024) typically fine-tune smaller models such as BERT (Devlin et al., 2019) or XLM-RoBERTa (Conneau et al., 2020) with large-scale contrastive learning. In contrast, instruction-based text embedding models are designed to use the target text along with a natural language instructions. Moreover, while LLMs are originally trained using causal attention, instruction-based approaches often incorporate additional modifications to enhance contextual understanding (Springer et al., 2024; Jiang et al., 2024; Lei et al., 2024). Notably, one common modification is the use of bidirectional attention (BehnamGhader et al., 2024; Lee et al., 2024a; de Souza P. Moreira et al., 2024). Although these models aim to generate task-specific embeddings to improve performance, whether they are truly capable of doing so and why performance improves have remained unclear. Our study is the first to qualitatively examine what embeddings prompt-based text embedding models produce and how their properties differ across prompts.

4.2 Embedding Dimensionality Reduction

Several attempts to reduce the dimensionality of embeddings have long been explored, with traditional methods such as PCA (Abdi and Williams, 2010) and Isomap (Tenenbaum et al., 2000). Wang et al. (2023) investigated effective dimensionality reduction methods by training models while adjusting the pooler's output dimensionality for each reduced dimension. Recently, learning methods designed to support post-hoc dimensionality reduction of embeddings, such as Matryoshka Representation Learning (MRL) (Kusupati et al., 2022) and Espresso Sentence Embeddings (LI et al., 2025), have also emerged. These methods incorporate specialized mechanisms during and after training to obtain high-performance embeddings even after dimensionality reduction. In contrast, our study aims to demonstrate that prompt-based text embedding models can achieve high performance on certain tasks using a simple dimensionality reduction approach, without requiring additional training or specialized datasets, and to investigate the underlying factors responsible for this behavior.

4.3 Intrinsic Dimension

The intrinsic dimension is defined as the minimum number of dimensions required to represent the underlying structure of data representations without significant information loss, and various methods have been proposed for estimating intrinsic dimensions (Bruske and Sommer, 1998; Fukunaga and Olsen, 1971; Levina and Bickel, 2004; Facco et al., 2017). Although the use of ID estimation on text embeddings is not yet widespread, there has been work applying ID-based methods to tasks such as detecting AI-generated text (Tulchinskii et al., 2023), by estimating the ID on sets of word embeddings for each text document. Dinu et al. (2025) investigate the impact of the temperature parameter on model performance. They demonstrate that increasing the temperature reduces the intrinsic dimensionality and degrades retrieval performance. To address this, they propose temperature aggregation and specialization methods, which integrate multiple temperatures directly into the contrastive training objective to balance performance and compressibility. While Dinu et al. (2025) consider temperature variation on a single model trained without any task-specific prompts, our findings indicate that LLM-based embedding models inherently modulate intrinsic dimensionality via instructions.

4.4 Isotropy and Anisotropy

It is well established that the contextualized word embeddings of language models are anisotropic, meaning they are predominantly distributed along a limited sub space within the embedding space. In research on text embeddings, enhancing isotropy has been shown to improve performance on STS tasks (Mu and Viswanath, 2018; Li et al., 2020; Su et al., 2021; Huang et al., 2021; Yokoi et al., 2024). In particular, training embedding models using contrastive learning techniques has been found to improve isotropy, thereby enhancing overall embedding quality (Gao et al., 2021; Zhuo et al., 2023; Xiao et al., 2023). Moreover, methods employing text embedding models for information retrieval (Karpukhin et al., 2020) have also reported performance gains through improvements in isotropy (Kim et al., 2024).

While improving the isotropy of embeddings has long been regarded as a key factor in improving their quality, recent studies have indicated that improving isotropy is not universally beneficial across all tasks. Specifically, Ait-Saada and Nadif (2023) point out that enhancing isotropy does not necessarily lead to improved performance in clustering tasks, and Mickus et al. (2024) argue that there exists a trade-off between the properties desirable for classification and clustering tasks, as measured by silhouette scores (Rousseeuw, 1987), and those for isotropy, which is generally preferred in STS and retrieval tasks. These findings indicate that the optimal level of isotropy in text embeddings may vary depending on the task. Our research supports this claim and further suggests that recent models attempt to navigate this trade-off by adjusting embeddings to exhibit varying degrees of isotropy.

5 Conclusion and Future Work

We demonstrated that the high-dimensional embeddings produced by prompt-based text embedding models can maintain strong performance even after dimensionality reduction by simply retaining the first several dimensions. In particular, for classification and clustering tasks, we showed that even drastic dimensionality reduction to just a few dimensions still preserved sufficient performance. Through analyses using intrinsic dimensionality and IsoScore, we found that prompt-based text embedding models generate embeddings with varying degrees of redundancy depending on the prompt. Specifically, for classification and clustering tasks, embeddings exhibit lower intrinsic dimensionality and tend to be less isotropic, and that is, they have higher redundancy. In contrast, for tasks like retrieval and STS, where fine-grained similarity is critical, embeddings tend to have higher intrinsic dimensionality and are more isotropically distributed, and that is, they have lower redundancy.

In future work, developing methods to construct embeddings with properties suitable for each task would be beneficial. Specifically, in contrastive learning, the temperature parameter is known to influence isotropy, with lower temperatures leading to more isotropic embeddings and higher temperatures resulting in less isotropic ones (Wang and Liu, 2021). Additionally, exploring more effective dimensionality reduction techniques for text embeddings remains an important direction. Embeddings may contain certain crucial dimensions, and if these dimensions can be identified, it may enable more efficient dimensionality reduction.

Limitations

In our study, we demonstrated that instruction-based text embedding models produce embeddings with different levels of redundancy depending on the prompt. However, we have not yet clarified the underlying factors that contribute to this phenomenon.

Furthermore, we estimated intrinsic dimension and isotropy using English Wikipedia text but did not conduct a detailed analysis of how these values might vary depending on text length, domain, or differences across languages. Expanding the range of datasets and conducting a more comprehensive analysis of downstream task performance would provide a stronger validation of how prompt-based text embeddings behave across different prompts.

Acknowledgement

This work was partly supported by JSPS KAKENHI Grant Numbers 23KJ1134 and 24H00727. We would also like to thank Ryo Ueda of the University of Tokyo for his insightful comments and for discussing isotropy and intrinsic dimensionality.

References

Hervé Abdi and Lynne J. Williams. 2010. Principal component analysis. *WIREs Computational Statistics*, 2(4):433–459.

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

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

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval)*, pages 497–511.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Semantic Evaluation (SemEval)*, pages 385–393.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic Textual Similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 32–43.

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 (ACL)*, pages 1194–1203.

Alessio Ansuini, Alessandro Laio, Jakob H. Macke, and Davide Zoccolan. 2019. Intrinsic dimension of data representations in deep neural networks. In *Neural Information Processing Systems (NeurIPS)*.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *International Conference on Learning Representations (ICLR)*.

Akari Asai, Timo Schick, Patrick Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen-tau Yih. 2023. Task-aware Retrieval with Instructions. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3650–3675.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders. In *First Conference on Language Modeling (COLM)*.

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 (EMNLP)*, pages 632–642.

Jörg Bruske and Gerald Sommer. 1998. Intrinsic dimensionality estimation with optimally topology preserving maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):572–575.

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

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8440–8451.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 670–680.

Gabriel de Souza P. Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. 2024. NV-Retriever: Improving text embedding models with effective hard-negative mining. *arXiv:2407.15831*.

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 (NAACL)*, pages 4171–4186.

Georgiana Dinu, Corey Barrett, Yi Xiang, Miguel Romero Calvo, Anna Currey, and Xing Niu. 2025. Effective post-training embedding compression via temperature control in contrastive training. In *International Conference on Learning Representations (ICLR)*.

Kawin Ethayarajh. 2018. Unsupervised Random Walk Sentence Embeddings: A Strong but Simple Baseline. In *Proceedings of the Third Workshop on Representation Learning for NLP (RepL4NLP)*, pages 91–100.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. 2017. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7.

Keinosuke Fukunaga and David R. Olsen. 1971. An Algorithm for Finding Intrinsic Dimensionality of Data. *IEEE Transactions on Computers*, C-20(2):176–183.

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 (EMNLP)*, pages 6894–6910.

Gregor Geigle, Nils Reimers, Andreas Rücklé, and Iryna Gurevych. 2021. TWEAC: Transformer with Extendable QA Agent Classifiers. *arxiv:2104.07081*, arXiv:2104.07081.

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, pages 1265–1268. ACM.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*.

Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. WhiteningBERT: An Easy Unsupervised Sentence Embedding Approach. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 238–244.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *arxiv:2310.06825*.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. 2024. Scaling Sentence Embeddings with Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*.

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 3784–3803.

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.

Jaeyoung Kim, Dohyeon Lee, and Seung-won Hwang. 2024. HIL: Hybrid Isotropy Learning for Zero-shot Performance in Dense retrieval. In *Proceedings of the 2024 Conference of the North American Chapter* of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pages 7892–7903.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham M. Kakade, Prateek Jain, and Ali Farhadi. 2022. Matryoshka Representation Learning. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS)*.

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 (TACL)*, 7:452–466.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024a. NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models. *arXiv:2405.17428*.

Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernandez Abrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha Reddy Jonnalagadda, Ming-Wei Chang, and Iftekhar Naim. 2024b. Gecko: Versatile Text Embeddings Distilled from Large Language Models. *arXiv:2403.20327*.

Yibin Lei, Di Wu, Tianyi Zhou, Tao Shen, Yu Cao, Chongyang Tao, and Andrew Yates. 2024. Meta-Task Prompting Elicits Embeddings from Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 10141–10157.

Elizaveta Levina and Peter Bickel. 2004. Maximum Likelihood Estimation of Intrinsic Dimension. In *Advances in Neural Information Processing Systems (NIPS)*, volume 17.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the Sentence Embeddings from Pre-trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130.

Xianming LI, Zongxi Li, Jing Li, Haoran Xie, and Qing Li. 2025. ESE: Espresso Sentence Embeddings. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards General Text Embeddings with Multi-stage Contrastive Learning. *arXiv:2308.03281*.

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 (ACL-HLT)*, pages 142–150.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, pages 216–223.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text.

Leland McInnes, John Healy, and James Melville. 2020. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv:1802.03426*.

Timothee Mickus, Stig-Arne Grönroos, and Joseph Attieh. 2024. Isotropy, Clusters, and Classifiers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 75–84.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-Top: Simple and Effective Postprocessing for Word Representations. In *International Conference on Learning Representations (ICLR)*.

Niklas Muennighoff. 2022. SGPT: GPT Sentence Embeddings for Semantic Search. *arXiv:2202.08904*.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative Representational Instruction Tuning. *arXiv:2402.09906*.

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 (EACL)*, pages 2014–2037.

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. *CoRR*, abs/1611.09268.

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.

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 (EMNLP)*, pages 9844–9855.

Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. 2024. Nomic Embed: Training a Reproducible Long Context Text Embedder. *arXiv:2402.01613*.

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 (EMNLP)*, pages 7092–7108.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. GPT-4 Technical Report. *arXiv:2303.08774*.

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.

Andrew Rosenberg and Julia Hirschberg. 2007. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420.

Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

William Rudman and Carsten Eickhoff. 2024. Stable Anisotropic Regularization. In *The Twelfth International Conference on Learning Representations (ICLR)*.

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.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 440–450.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. 2024. Repetition Improves Language Model Embeddings. *arXiv:2402.15449*.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One Embedder, Any Task: Instruction-Finetuned Text Embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening Sentence Representations for Better Semantics and Faster Retrieval. *arXiv:2103.15316*.

Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. 2000. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323.

Hayato Tsukagoshi, Ryohei Sasano, and Koichi Takeda. 2021. DefSent: Sentence Embeddings using Definition Sentences. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 411–418.

Eduard Tulchinskii, Kristian Kuznetsov, Kushnareva Laida, Daniil Cherniavskii, Sergey Nikolenko, Evgeny Burnaev, Serguei Barannikov, and Irina Piontkovskaya. 2023. Intrinsic Dimension Estimation for Robust Detection of AI-Generated Texts. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*.

Laurens van der Maaten and Geoffrey E. Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Feng Wang and Huaping Liu. 2021. Understanding the Behaviour of Contrastive Loss. pages 2495–2504.

Hongwei Wang, Hongming Zhang, and Dong Yu. 2023. On the Dimensionality of Sentence Embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10344–10354.

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:2212.03533*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024a. Improving Text Embeddings with Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 11897–11916.

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

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 (NAACL)*, pages 1112–1122.

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.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2024. C-Pack: Packaged Resources To Advance General Chinese Embedding. In *The 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 641–649.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 Technical Report. *arxiv:2407.10671*.

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 (EMNLP)*, pages 2369–2380.

Chihiro Yano, Akihiko Fukuchi, Shoko Fukasawa, Hideyuki Tachibana, and Yotaro Watanabe. 2024. Multilingual Sentence-T5: Scalable Sentence Encoders for Multilingual Applications. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*, pages 11849–11858.

Sho Yokoi, Han Bao, Hiroto Kurita, and Hidetoshi Shimodaira. 2024. Zipfian whitening. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 122259–122291.

Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2022. Making a MIRACL: Multilingual Information Retrieval Across a Continuum of Languages. *arXiv:2210.09984*.

Wenjie Zhuo, Yifan Sun, Xiaohan Wang, Linchao Zhu, and Yi Yang. 2023. WhitenedCSE: Whitening-based Contrastive Learning of Sentence Embeddings. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 12135–12148.

A Details of Evaluation Tasks and Prompts

Table 3 presents the tasks from MTEB used in our experiments, along with the instructions employed for each task when using instruction-based text embedding models.

B Model Details Used in Our Experiments

Table 4 lists the models used in the evaluation experiments. We provide detailed descriptions of each model below.

Instruction-Based Text Embedding Models We evaluate several models that have demonstrated high performance on the MTEB. In particular, we consider the following four models, three of which are LLM-based, while the other based on XLM-RoBERTa (Conneau et al., 2020):

- **gte-Qwen2**[9]: A fine-tuned version of Qwen2 7B (Yang et al., 2024) fine-tuned for text embeddings. This model replaces its original causal attention with bidirectional attention and is further trained on diverse multilingual datasets. The model comprises 7.6B parameters and produces embeddings with 3,584 dimensions.

- **E5-mistral** (Wang et al., 2024a): A fine-tuned variant of Mistral 7B (Jiang et al., 2023) that leverages synthetic data generated by high-performant LLMs, such as GPT-4 (OpenAI et al., 2024). It is a pioneering model in LLM and instruction-based text embeddings, demonstrating that the model more accurately captures the objectives of the embedding task and yields better embeddings. The model comprises 7.1B parameters and produces embeddings with 4,096 dimensions.

- **SFR-Embedding-2_R (SFR-2)**[10]: An enhanced version of E5-mistral which is further fine-tuned using LoRA (Hu et al., 2022). The model comprises 7.1B parameters and produces embeddings with 4,096 dimensions.

- **mE5-large-inst** (Wang et al., 2024b): A multilingual and instruction-based version of E5 (Wang et al., 2022) fine-tuned on the same datasets as

[9]https://huggingface.co/Alibaba-NLP/gte-Qwen2-7B-instruct

[10]https://huggingface.co/Salesforce/SFR-Embedding-2_R

Task	Instruction
AmazonCounterfactualClassification	Classify a given Amazon customer review text as either counterfactual or not-counterfactual
AmazonPolarityClassification	Classify Amazon reviews into positive or negative sentiment
AmazonReviewsClassification	Classify the given Amazon review into its appropriate rating category
ImdbClassification	Classify the sentiment expressed in the given movie review text from the IMDB dataset
ToxicConversationsClassification	Classify the given comments as either toxic or not toxic
ArxivClusteringS2S	Identify the main and secondary category of Arxiv papers based on the titles
RedditClustering	Identify the topic or theme of Reddit posts based on the titles
StackExchangeClustering	Identify the topic or theme of StackExchange posts based on the titles
MIRACLRetrievalHardNegatives	Given a question, retrieve Wikipedia passages that answer the question
QuoraRetrievalHardNegatives	Given a question, retrieve questions that are semantically equivalent to the given question
HotpotQAHardNegatives	Given a multi-hop question, retrieve documents that can help answer the question
DBPediaHardNegatives	Given a query, retrieve relevant entity descriptions from DBPedia
NQHardNegatives	Given a question, retrieve Wikipedia passages that answer the question
MSMARCOHardNegatives	Given a web search query, retrieve relevant passages that answer the query
STS-12	Retrieve semantically similar text
STS-13	Retrieve semantically similar text
STS-14	Retrieve semantically similar text
STS-15	Retrieve semantically similar text
STS-16	Retrieve semantically similar text
STS-Benchmark	Retrieve semantically similar text
SICK-R	Retrieve semantically similar text

Table 3: Evaluation tasks and their corresponding instructions.

Model	HuggingFace	Prompt	Dim.	#Params
gte-Qwen2	Alibaba-NLP/gte-Qwen2-7B-instruct	Instruction	3584	7.61B
E5-mistral	intfloat/E5-mistral-7b-instruct	Instruction	4096	7.11B
SFR-2	Salesforce/SFR-Embedding-2_R	Instruction	4096	7.11B
mE5-large-inst	intfloat/multilingual-e5-large-instruct	Instruction	1024	560M
nomic	nomic-ai/nomic-embed-text-v1.5	Prefix (five types)	768	137M
E5-small	intfloat/e5-small-v2	Prefix (two types)	384	33M
E5-large	intfloat/e5-large-v2	Prefix (two types)	1024	335M
Unsup-SimCSE	princeton-nlp/unsup-simcse-bert-large-uncased	N/A	1024	335M

Table 4: Details of each model.

E5-mistral. Unlike the aforementioned models, mE5-large-inst is derived from XLM-RoBERTa-large (Conneau et al., 2020). The model comprises 560M parameters and produces embeddings with 1,024 dimensions.

Each model generates task-specific embeddings by incorporating tailored instructions into the input text. Specifically, task instructions are prepended to the input texts prior, thereby enabling effective adaptation to a wide range of downstream tasks. The instructions used are identical to those used in previous studies (Wang et al., 2024a; Lee et al., 2024a), and the ones for each task are provided in Appendix A.

Other Text Embedding Models Small-scale prefix-based models are highly valuable in practice, and understanding how they differ from instruction-based models is crucial; therefore, in addition to the instruction-based embedding model, we considered the following four models:

- **Unsup-SimCSE**: SimCSE (Gao et al., 2021) is a method for fine-tuning language models into text embedding models using contrastive learning. In our experiments, we employ the BERT-large model fine-tuned on one million English Wikipedia sentences.[11] This model consists of 335M parameters and outputs embeddings with 1,024 dimensions. Notably, Unsup-SimCSE does not rely on prompts.

- **E5 (E5-large and E5-small)**: E5 (Wang et al., 2022) is a prefix-based text embedding model fine-tuned with large-scale contrastive learning on diverse datasets. During contrastive learning, E5 appends prefixes such as `query:` and `passage:` to the input text, thereby enabling the effective computation of asymmetric similarities between retrieval queries and documents. Although several model sizes are available, in

[11] https://huggingface.co/princeton-nlp/unsup-simcse-bert-large-uncased

Dim.	2	4	8	16	32	64	128	256	384	512	768	896	1024	3584
							Classification							
First	76.34	83.39	85.73	86.91	87.16	87.11	86.85	87.12	87.10	87.14	87.11	87.18	87.26	87.15
Random	72.87	81.64	85.42	86.21	86.79	86.84	86.79	87.25	86.98	87.00	87.15	87.14	87.16	-
PCA	84.86	85.05	85.22	85.25	85.31	85.36	85.40	85.42	85.43	85.43	85.44	85.44	85.44	-
UMAP	82.99	84.59	84.97	84.50	84.16	83.58	83.33	83.19	83.12	83.28	83.18	83.17	83.29	-
Isomap	83.61	85.19	85.25	85.30	85.36	85.45	85.55	85.66	85.69	85.72	85.74	85.75	85.76	-
							Clustering							
First	22.88	41.55	56.13	62.44	66.08	67.43	67.52	67.93	68.06	68.06	68.32	68.33	68.15	68.40
Random	24.70	41.41	55.33	62.71	65.89	67.13	67.44	67.89	68.12	68.15	68.25	68.29	68.26	-
PCA	38.64	55.63	62.93	66.00	67.81	68.49	68.40	68.38	68.43	68.33	68.37	68.44	68.48	-
UMAP	53.16	64.16	65.25	65.45	65.49	65.56	65.49	65.50	65.46	65.37	65.45	65.37	65.32	-
Isomap	42.50	58.68	63.72	65.51	66.24	66.11	66.09	66.00	66.00	65.82	65.61	65.82	65.83	-
							Retrieval							
First	0.39	0.57	4.78	20.34	41.43	56.08	62.17	65.72	66.81	67.64	67.98	68.15	68.33	69.22
Random	0.34	0.88	5.02	20.02	40.41	54.98	61.80	65.52	66.88	67.43	68.10	68.31	68.31	-
PCA	1.45	5.00	15.34	31.11	45.49	56.23	62.95	66.30	67.32	68.00	68.61	68.89	68.95	-
UMAP	2.70	5.81	5.46	5.21	5.01	4.60	3.96	3.60	3.39	3.13	3.04	2.83	2.53	-
Isomap	1.45	9.53	20.89	24.99	27.03	27.92	28.54	28.78	28.91	29.08	29.32	29.31	29.58	-
							STS							
First	38.43	54.29	66.49	73.65	78.19	80.15	81.52	83.04	82.82	83.44	84.00	84.18	84.18	84.76
Random	35.18	51.33	63.48	71.83	76.79	80.02	82.42	83.21	83.80	84.03	84.38	84.43	84.56	-
PCA	33.44	50.71	61.85	69.08	75.22	80.17	83.45	84.95	85.23	85.31	85.28	85.24	85.19	-
UMAP	53.20	63.40	65.64	65.38	65.15	65.54	65.48	65.71	65.90	65.47	65.37	65.38	65.95	-
Isomap	45.40	62.08	69.60	73.85	75.35	76.28	76.71	76.88	76.83	76.73	76.50	76.46	76.45	-

Table 5: Performance of gte-Qwen2 under various dimensionality reduction methods.

this study we use the small model (E5-small)[12] comprises 33M parameters and its embedding dimension is 384 and the large model (E5-large)[13] comprises 335M parameters and its embedding dimension is 1,024.

- **nomic**: Nomic further develops the prefix approach used in E5 by employing five distinct prefixes tailored for different tasks. Specifically, the prefix `search_query:` is used for queries of retrieval, `search_document:` for documents of retrieval, `classification:` for classification, `clustering:` for clustering, and, for tasks such as STS, where the semantic content of the text is to be embedded, it will be an empty string (i.e., no prefix). The model comprises 137M parameters and its embedding dimension is 768.[14]

C Experimental Results of Other Dimensionality Reduction Methods

Taking the first d dimensions (First) aside, we evaluate four alternative dimensionality reduction methods: (1) Random, which selects random embedding coordinates; (2) PCA (Abdi and Williams, 2010); (3) UMAP (McInnes et al., 2020); and (4) Isomap (Tenenbaum et al., 2000). First, the Random and PCA methods perform only linear transformations, whereas UMAP and Isomap implement nonlinear transformations. The performance of each method are shown in Table 5. For the Random method, we fix the indices of selected dimensions prior to each task evaluation to ensure that the same subset of embedding coordinates is used consistently across runs. Although t-SNE (van der Maaten and Hinton, 2008) is a well-known dimensionality reduction method, we did not employ it because it becomes computationally infeasible for projections into high-dimensional spaces: t-SNE incurs a per-iteration time complexity of $O(N^2 d)$, where N denotes the number of data points. We use gte-Qwen2 for the embedding model. All remaining settings are identical to those specified in Section 2.

The results indicate that, despite its simplicity, the First method achieves performance comparable to or better than the other methods when using 8 dimensions for classification, 16 for STS, 32 for clustering, and 64 for retrieval. Moreover, more computationally intensive methods do not necessarily yield improved results and may even degrade performance in higher-dimensional settings.

[12] https://huggingface.co/intfloat/e5-small-v2
[13] https://huggingface.co/intfloat/e5-large-v2
[14] https://huggingface.co/nomic-ai/nomic-embed-text-v1.5

Classification Tasks Across all dimensionality reduction methods, performance decreases gradually as the number of dimensions is reduced. At very low dimensions (2 and 4), PCA, UMAP, and Isomap outperform the other approaches; however, for dimensions ≥ 8, the First and Random methods consistently surpass them. We hypothesize that PCA, UMAP, and Isomap transformations can distort the geometric structures that are critical for classification, whereas retaining the leading dimensions better preserves these intrinsic structures.

Clustering Tasks Similarly, clustering performance declines gradually for all methods. UMAP achieves strong performance at extremely low dimensions. PCA matches the First method across all evaluated dimensions. The performance of the other methods deteriorates at higher dimensions.

Retrieval and STS Tasks Consistent with the observations in Section 2, all dimensionality reduction methods exhibit a rapid decline in retrieval performance as the number of dimensions decreases. PCA slightly outperforms the First method in retrieval tasks; however, the improvement is marginal. UMAP and Isomap fail to achieve competitive performance at relatively higher dimensions, likely due to distortions introduced by their nonlinear transformations.