LDIR: Low-Dimensional Dense and Interpretable Text Embeddings with Relative Representations

Yile Wang, Zhanyu Shen, Hui Huang*
College of Computer Science and Software Engineering, Shenzhen University
wangyile@szu.edu.cn, zhanyushen.s@gmail.com, hhzhiyan@gmail.com

Abstract

Semantic text representation is a fundamental task in the field of natural language processing. Existing text embedding (e.g., SimCSE and LLM2Vec) have demonstrated excellent performance, but the values of each dimension are difficult to trace and interpret. Bag-of-words, as classic sparse interpretable embeddings, suffers from poor performance. Recently, Benara et al. (2024) propose interpretable text embeddings using large language models, which forms "0/1" embeddings based on responses to a series of questions. These interpretable text embeddings are typically high-dimensional (larger than 10,000). In this work, we propose Low-dimensional (lower than 500) Dense and Interpretable text embeddings with Relative representations (LDIR). The numerical values of its dimensions indicate semantic relatedness to different anchor texts through farthest point sampling, offering both semantic representation as well as a certain level of traceability and interpretability. We validate LDIR on multiple semantic textual similarity, retrieval, and clustering tasks. Extensive experimental results show that LDIR performs close to the black-box baseline models and outperforms the interpretable embeddings baselines with much fewer dimensions. Code is available at https://github.com/szu-tera/LDIR.

1 Introduction

Text embedding is an important technique for natural language processing by transforming textual data into numerical representations that capture semantics. The embeddings encode the meaning of contexts in a vector space where similar texts are close together in the representation space, playing vital roles in multiple tasks such as semantic textual similarity (Agirre et al., 2012), information retrieval (Karpukhin et al., 2020), and retrieval-augmented generation (Lewis et al., 2020).

Embeddings	**Type**	**Interp.**	**Examples**
SimCSE LLM2Vec	Dense Emb (768∼4,096 dim.)	✗	0.71, -0.03, ..., 0.13 (uninterpretable)
Bag-of-Words	Sparse Emb (∼30,000 dim)	✓	12, 5, ..., 0 (occurrence of <word>$_i$)
QAEmb-MBQA CQG-MBQA	0/1 Emb (∼10,000 dim.)	✓	1, 0, ..., 1 ("yes/no" to <question>$_i$)
LDIR (Ours)	**Dense Emb** **(∼500 dim.)**	✓	**0.16, 0.83, ..., 0.35** **(relatedness to <anchor>$_i$)**

Table 1: Comparing existing text embedding with LDIR. In particular, our text embedding is relatively low-dimensional and dense, while maintaining interpretable.

Seminal text embedding models mainly fall into two categories. The first is based on lightweight pre-trained models such as BERT (Devlin et al., 2019), including Sentence-BERT (Reimers and Gurevych, 2019), SimCSE (Gao et al., 2021), etc. The second is based on current large language models such as LLaMA-3 (Dubey et al., 2024), including LLM2Vec (BehnamGhader et al., 2024), AngIE (Li and Li, 2024), etc. The text embeddings usually have 768 to 4096 dimensions correlated with the hidden size of the model and perform well across tasks. However, the values on each dimension are difficult to trace and interpret directly.

In contrast, bag-of-words is a classical sparse and interpretable text representation where each dimension's value represents the frequency of a specific word. However, its representational capability is limited. Recent QAEmb-MBQA (Benara et al., 2024) and CQG-MBQA (Sun et al., 2025) generate interpretable text embeddings based on large language models, where the value on each dimension is restricted to "0/1", reflecting "yes/no" answers to different crafted questions, as shown in Table 1. These works pave the way for new developments in interpretable text embedding.

In the aforementioned interpretable text embeddings, the value on each dimension are explicitly defined. However, due to the limitations of their sparse or "0/1" representations, they often require

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2025, pages 14397–14409

a high dimensionality. For example, bag-of-words is associated with the word vocabulary size, which typically reaches around 30K sizes in current models. QAEmb-MBQA and CQG-MBQA rely on designing a large number of questions, usually around 10K. In this work, we aim to mitigate the limitations of sparse or "0/1" representation and the requirement of high dimensionality.

Inspired by Moschella et al. (2023), we propose building low-dimensional, dense, and interpretable text embeddings with relative representations (LDIR). Specifically, the value on each dimension represents the relatedness to different "anchor texts" by automatic farthest point sampling, which can be floating-point numbers instead of "0/1", thereby leading a overall flexible and dense embeddings. We hope the dimensionality of vectors can be significantly reduced in this way, achieving a balance among semantic expressiveness, representational efficiency, and interpretability.

We conduct extensive experiments to validate the semantic expressiveness of LDIR, including evaluations on seven semantic textual similarity tasks, six retrieval tasks, and seven clustering tasks. Experimental results show that our embeddings, with only 500 embedding dimensions, achieve comparable performance to black-box embeddings, and outperforms interpretable embedding baselines of QAEmb-MBQA and CQG-MBQA.

2 Related Work

Semantic Text Embedding. For pre-trained neural model based text embeddings, Sentence-BERT (Reimers and Gurevych, 2019) and SimCSE (Gao et al., 2021) use siamese network or contrastive learning to improve the semantic representations of BERT (Devlin et al., 2019). For large language model based text embeddings, MetaEOL (Lei et al., 2024) elicit the text embeddings from a single-token through prompting LLaMA (Touvron et al., 2023). AngIE (Li and Li, 2024) optimizes angle differences instead of cosine similarity for better modeling semantics. LLM2Vec (BehnamGhader et al., 2024) enables bidirectional attention and avoid the limitations of decoder-only models in encoding texts. NV-Embed (Lee et al., 2025) propose latent attention layer to enhance the general-purpose embedding tasks of decoder-only models. Although these methods produce dense text embedding and show strong performance on benchmarks, however, it's difficult to interpret each dimension of the resulted dense representations.

Interpretable Text Embedding. Bag-of-words (BoW) is one of the most classic and interpretable text embedding, where each dimension represents the frequency of occurrence for each word in the vocabulary. However, its representation capacity is largely limited. LISA (Patel et al., 2023) distills linguistically interpretable style attribute embeddings for author style representation, which is not general semantics oriented. QAEmb-MBQA (Benara et al., 2024) proposes designing multiple questions and use the "yes/no" answers by language models as "0/1" embeddings for semantic representations. CQG-MBQA (Sun et al., 2025) further improves the question generation module and reduce the costs for obtaining the "0/1" embeddings. Unlike these approaches, we develop low-dimensional dense text embeddings which can improve the general semantic expressiveness while preserving a certain of interpretability.

Relative Representation. Moschella et al. (2023) observe that the latent similarity between each sample and a fixed set of anchors in neural networks shows invariance for different training settings. They call such latent similarity as relative representation and verify the findings on the space of word embeddings (Mikolov, 2013; Bojanowski et al., 2017) or model stitching (Bansal et al., 2021). Norelli et al. (2023) use relative representations to align images and text for zero-shot classification. Maniparambil et al. (2024) also leverage such cross-modal alignment for caption matching and retrieval tasks. Overall, the above studies primarily illustrate the potential of relative representation for feature alignment, rather than utilizing it for sentence-level semantic representations.

3 Method

3.1 QA-Embedding Baseline

Formally, we define the semantic text embedding $e(t)$ of a given a text t as:

$$e(t) = \mathcal{E}(t) = [v_1, v_2, ..., v_d], \quad (1)$$

where $\mathcal{E}$ is an embedding model, $v_1, v_2, ..., v_d$ are the values in each dimension. For neural model $\mathcal{E}$, the embedding $[v_1, v_2, ..., v_d]$ generally encode some high-level semantic features. However, it is difficult to interpret the specific meanings of each value v_i directly.

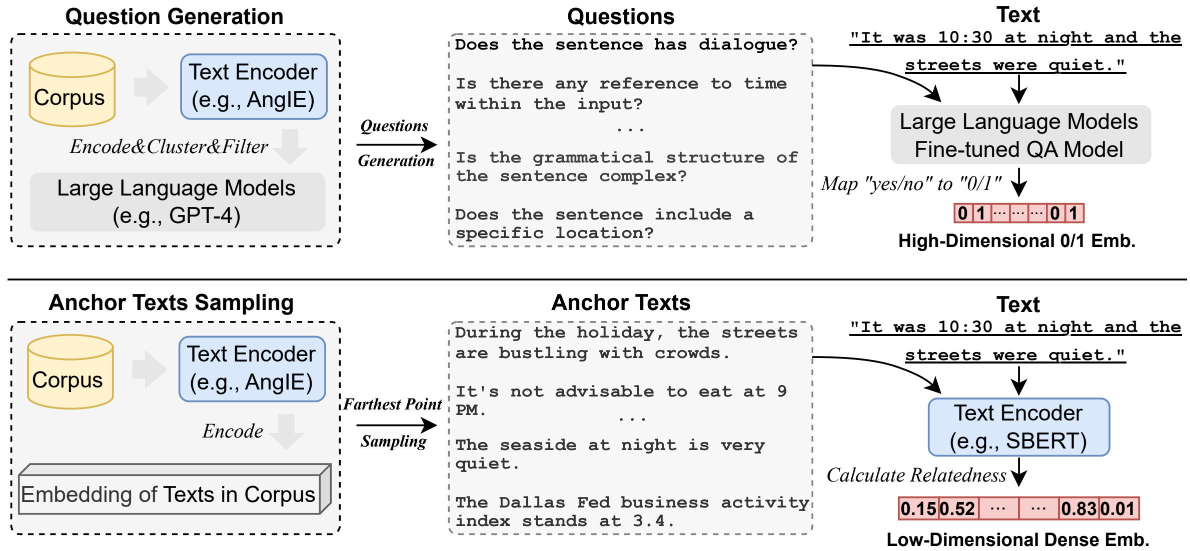

Figure 1: Comparison between baselines (top) and our method (bottom). Benara et al. (2024) and Sun et al. (2025) uses the generated questions and "yes/no" answers to build high-dimensional "0/1" text embeddings. We use anchor texts via farthest point sampling to calculate the relatedness and build low-dimensional dense text embeddings.

Benara et al. (2024) propose generating interpretable QA-Embedding to make the values in each dimension meaningful. In particular, they first generate k questions $q_1, q_2, ..., q_k$ by leveraging GPT-4 with specific prompts, then obtain the k "yes" or "no" responses $\{r_j\}_{j=1}^k$ by using large language models $\mathcal{M}$ (e.g., GPT-4 or LLaMA-3):

$$r_1, ..., r_k = \mathcal{M}(q_1 \oplus t), ..., \mathcal{M}(q_k \oplus t), \qquad (2)$$

where $\oplus$ indicates concatenation operation, and the answer r_j is either "yes" or "no" to the question q_j.

The interpretable text embedding is defined as the "0/1" sequence of according to the answers:

$$\begin{aligned} e_{0/1}^{\text{interp}}(t) &= [\hat{r}_1, \hat{r}_2, ..., \hat{r}_k], \\ \hat{r}_j &= \begin{cases} 1, & \text{if } r_j = \mathcal{M}(q_j \oplus t) \text{ is "yes"}, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \qquad (3)$$

Sun et al. (2025) further improves the QA-Embedding by reducing the costs of calling LLMs through applying external trainable question generation module. We show an example of QA-Embedding in the top of Figure 1.

Both Benara et al. (2024) and Sun et al. (2025) rely on high-quality questions $\{q_j\}_{j=1}^k$ generated via GPT-4 or trained models, followed by extra filtering. On the other hand, since $\hat{r}_j$ can only be 0 or 1, the number of questions is extremely large, often reaching 9000∼10000 (i.e., the number k).

3.2 "0/1" Embedding to Dense Embedding

Although the above values $\{\hat{r}_j\}_{j=1}^k$ of the embedding $e^{\text{interp}}(t)$ is interpretable, the expressive power is greatly limited because all values are either 0 or 1. Therefore, we consider using dense representations with floating-point number of each dimension. The overview of our dense and interpretable embedding is illustrated in the bottom of Figure 1.

Instead of generating k questions $q_1, q_2, ..., q_k$, we first select n representative texts $a_1, a_2, ..., a_n$ which we called "anchor texts" (detailed in the following subsection). Then, for each anchor text a_j, we compute the relatedness between a_j and t.

$$s_j = \textsc{Rel}(a_j, t), \qquad (4)$$

where $\textsc{Rel}(a_j, t)$ indicates the relatedness score of a_j and t, which can be computed through general encoder or trained text encoders using widely used cosine similarity (Mikolov, 2013; Reimers and Gurevych, 2019; Ethayarajh, 2019):

$$\textsc{Rel}(a_j, t) = \frac{\textsc{Enc}(a_j) \cdot \textsc{Enc}(t)}{||\textsc{Enc}(a_j)|| \cdot ||\textsc{Enc}(t)||}, \qquad (5)$$

where $\textsc{Enc}$ can be an pre-trained encoder such as SimCSE (Gao et al., 2021), ModernBERT (Warner et al., 2024), and AnglE (Li and Li, 2024), without further fine-tuning.

After obtaining the relatedness scores $\{s_j\}_{j=1}^n$, our dense and interpretable embeddings is:

$$\begin{aligned} e_{\text{dense}}^{\text{interp}}(t) &= [s_1, ..., s_n] \\ &= [\textsc{Rel}(a_1, t), ..., \textsc{Rel}(a_n, t)]. \end{aligned} \qquad (6)$$

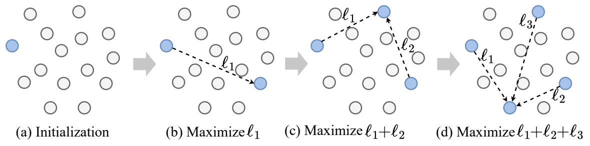

Figure 2: Example of extracting four anchor texts using farthest point sampling. Each dot represents a text embedding through an encoder, with the blue dots indicating the extracted anchor texts.

Comparing Eq. 6 with Eq. 3, our embeddings $e_{\text{dense}}^{\text{interp}}(t)$ are represented based on floating-point numbers, which can alleviate the requirement for high dimensionality. In practice, we find that 200∼500 (i.e., the number n) anchor texts are enough to achieve better performance.

3.3 Anchor Texts Selection

In contrast to other interpretable embedding methods that rely on the extensive general knowledge of large language models (e.g., GPT-4) to derive and filter explanatory questions, we propose to automatically sample anchor texts from the corpus without generation and filtering. Selecting representative texts $a_1, a_2, ..., a_n$ as anchor texts are important. For example, if all the anchor texts are similar to each other, then values in the final embeddings in Eq. 6 will also be close, thus the representativeness of overall embedding is limited. In practice, we use the farthest point sampling algorithm to select the representative anchor texts and compare with other sampling methods such as uniform sampling and K-Means in Section 5.1.

Farthest point sampling (FPS) is a widely used method that identifies representative subset within the complete feature space. In our workflow, the generation process begins by applying an encoder to the corpus texts, producing N embeddings $\{e_1, e_2, ..., e_N\}$. Then a subset $\{\hat{e}_1, \hat{e}_2, ..., \hat{e}_n\}$ is sampled gradually such that the distance between any two embedding vectors $(\hat{e}_i, \hat{e}_j)$ are farther apart than that of any embedding vector pair in the remaining set $\{e_1, e_2, ..., e_{N-n}\}$. Therefore, the text corresponding to the collected subset of embeddings constitutes the anchor texts $a_1, a_2, ..., a_n$ required. An example of FPS procedure is shown in Figure 2. Benara et al. (2024) uses AngIE (Li and Li, 2024) to encode texts in corpus for generating questions by GPT-4. We follow them and set AngIE as the text encoder.

Embeddings	Type	Dim.↓	†Ext. Costs↓	Interp.↑	Perf.↑
SimCSE LLM2Vec	Dense	Medium	Medium	Low	High
Bag-of-Words	Sparse	Large	Low	High	Low
QAEmb-MBQA CQG-MBQA	0/1 Emb.	Large	High	High	Medium
LDIR (Ours)	Dense	Small	Low	Medium	High

Table 2: Overall comparison between existing embeddings and LDIR. †: The external costs indicate the costs for external training or calling large language models.

3.4 Overall Comparison

We make an overall comparison to existing embedding methods in Table 2. Comparing from different perspectives, we achieve **low-dimensional and dense embeddings with low costs**. Specifically, SimCSE and LLM2Vec require additional contrastive learning to fine-tune models, QAEmb-MBQA and CQG-MBQA require extra calls on GPT-4 or training question generation modules. As comparison, LDIR primarily employs a automatically farthest point sampling strategy to extract anchor texts, and the calculation of embeddings does not require prompting large language models.

We also notice a **trade-off between interpretability and performance** across embeddings. For example, SimCSE and LLM2Vec exhibit strong semantic representation capabilities, but their interpretability is relatively low. On the other hand, bag-of-words offers high interpretability but performs poorly. Compared to QAEmb-MBQA and CQG-MBQA, LDIR sacrifices some interpretability (as the overall semantic relatedness is less direct and explicit than responses to questions) but achieves better performance, comparable to embeddings like SimCSE (see experimental results in Section 4.2). In Section 5.3, we further discuss how to obtain fine-grained relatedness to enhance the interpretability of our approach.

4 Experiments

4.1 Settings

Corpus. We use the same texts resource MEDI2[1] and pre-processing steps (merging and filtering) by Sun et al. (2025), which results in a final corpus of approximately 6.8 million sentences for task-agnostic anchor texts extraction.

Tasks and Datasets. For semantic textual similarity (STS) tasks, we evaluate on SemEval STS

[1] https://huggingface.co/datasets/GritLM/MEDI2

Model	Dim.	Type	Spearman Correlation (STS)							
			STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
(Black-Box Embeddings)										
$\text{BERT}_{\text{base}}$ (Devlin et al., 2019)	768	Dense	38.78	57.98	57.98	63.15	61.06	46.35	58.40	54.81
GloVe (Pennington et al., 2014)	300	Dense	54.64	69.16	60.81	72.31	65.34	61.54	55.43	62.74
USE (Cer et al., 2018)	512	Dense	64.49	67.80	64.61	76.83	73.18	74.92	76.69	71.22
LLM2Vec (BehnamGhader et al., 2024)	4,096	Dense	61.60	79.71	72.11	82.18	79.41	77.44	72.16	74.94
$\text{SimCSE}_{\text{unsup}}$ (Gao et al., 2021)	768	Dense	66.05	81.49	73.61	79.72	78.12	76.52	72.24	75.39
$\text{SBERT}_{\text{ori}}$ (Reimers and Gurevych, 2019)	768	Dense	74.53	77.00	73.18	81.85	76.82	79.10	74.29	76.68
$\text{SimCSE}_{\text{sup}}$ (Gao et al., 2021)	768	Dense	75.30	84.67	80.19	85.40	80.82	84.25	68.38	79.86
WhitenedCSE (Zhuo et al., 2023)	768	Dense	74.65	85.79	77.49	84.71	80.33	81.48	75.34	79.97
$\text{SBERT}_{\text{new}}$ (Reimers and Gurevych, 2019)	768	Dense	73.08	82.13	76.73	85.58	80.23	83.09	79.32	80.02
text-embedding-ada-002 (OpenAI, 2022)	1,536	Dense	72.84	86.10	81.15	88.49	85.08	83.56	79.00	82.31
$\text{ModernBERT}_{\text{emb}}$ (Warner et al., 2024)	1,024	Dense	80.67	87.87	83.80	88.59	86.82	87.40	80.31	85.07
AngIE (Li and Li, 2024)	1,024	Dense	79.09	89.62	85.02	89.51	86.61	89.06	82.62	85.93
(Interpretable Embeddings)										
Bag-of-Words (with BERT vocabulary)	28,996	Sparse	44.75	52.06	54.78	68.65	60.59	54.85	57.87	56.22
QAEmb-MBQA (Benara et al., 2024)	10,654	0/1 Emb.	59.40	63.19	57.68	69.29	63.18	71.33	72.33	65.20
CQG-MBQA (Sun et al., 2025)	9,614	0/1 Emb.	69.21	80.19	73.91	80.66	78.30	82.69	78.21	77.60
LDIR (with $\text{SBERT}_{\text{new}}$)	200	Dense	74.44	77.73	73.58	81.93	77.98	80.82	78.42	77.84
LDIR (with $\text{SBERT}_{\text{new}}$)	500	Dense	72.51	79.92	74.93	82.93	78.44	81.88	79.34	78.55
LDIR (with $\text{ModernBERT}_{\text{emb}}$)	200	Dense	72.99	79.63	75.09	82.52	79.93	82.61	78.43	78.74
LDIR (with $\text{ModernBERT}_{\text{emb}}$)	500	Dense	72.30	81.97	75.48	82.86	79.88	83.22	80.72	79.49
LDIR (with AngIE)	200	Dense	**79.28**	82.16	**80.96**	84.64	**83.68**	85.32	77.92	81.99
LDIR (with AngIE)	500	Dense	78.85	**84.35**	80.93	**84.79**	83.61	**86.31**	**80.85**	**82.82**

Table 3: Comparison between baselines and LDIR on semantic textual similarity (STS) tasks. Among the interpretable embeddings, the best results in each column are **in bold**, and the second-best results are underlined.

tasks 2012-2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS Benchmark (Cer et al., 2017), and SICK-Relatedness (Marelli et al., 2014).

For retrieval tasks, we use 1% of the samples from the MS MARCO (Bajaj et al., 2016) development set, ArguAna (Wachsmuth et al., 2018), FiQA-2018 (FQA; Maia et al., 2018), NFCorpus (NFC; Boteva et al., 2016), SCIDocs (Cohan et al., 2020), and SciFact (Wadden et al., 2020).

For clustering tasks, we use TwentyNewsgroups (TNG), StackExchange (SE-P2P), Biorxiv (BR-P2P, BR-S2S), Medrxiv (MR-P2P, MR-S2S), and Reddit (RD-P2P) from massive text embedding benchmark (MTEB; Muennighoff et al., 2023).

Baselines. We use the baselines in Sun et al. (2025), including GloVe (Pennington et al., 2014), USE (Cer et al., 2018), BERT (Devlin et al., 2019), SBERT (Reimers and Gurevych, 2019), SimCSE (Gao et al., 2021), API-based text-embedding-ada-002 (OpenAI, 2022), Whitened-CSE (Zhuo et al., 2023), AngIE (Li and Li, 2024), Bag-of-Words, BM25 (Robertson et al., 2009), QAEmb-MBQA (Benara et al., 2024). We also use LLM2Vec (BehnamGhader et al., 2024), ModernBERT (Warner et al., 2024), and CQG-MBQA (Sun et al., 2025) as additional baselines.

Evaluations. We use the official evaluation suite of BEIR (Thakur et al., 2021) for MS MARCO and MTEB (Muennighoff et al., 2023) for other datasets. Spearman correlation, nDCG@10, and V-measure metrics are used as metrics for STS, retrieval, and clustering tasks, respectively.

4.2 Main Results

We apply different encoders ENC in Eq. 5 and sample different number of anchor texts ($n = 200$ and $n = 500$) for comparison.

Semantic Textual Similarity. The results are shown in Table 3. LDIR demonstrates significant performance improvements among interpretable text embedding models while achieving competitive results compared to state-of-the-art black-box models. For example, LDIR outperforms the best interpretable embeddings CQG-MBQA (82.82 vs. 77.60), and also surpasses black-box models including $\text{SBERT}_{\text{new}}$ (80.02) and OpenAI's text-embedding-ada-002 (82.31).

Comparing different dimensional settings, the 500-dimensional embeddings perform slightly better than the 200-dimensional ones. Among different backbone encoder models, AngIE gives the best results, while SBERT performs the worst, reflecting their differences in semantic encoding capabilities.

Retrieval. The results are shown in Table 4. LDIR achieves a nDCG@10 score of 46.68 with AngIE encoder and 500 dimensions, outperforming the traditional retrieval algorithm BM25 (43.23) and significantly outperforms the interpretable embed-

Model	Dim.	Type	nDCG@10 (Retrieval)						
			MS MARCO	ArguAna	FQA	NFC	SCIDocs	SciFact	Avg.
(Black-Box Embeddings)									
$\text{BERT}_{\text{base}}$ (Devlin et al., 2019)	768	Dense	16.86	28.29	2.19	4.30	2.82	13.34	11.30
$\text{SimCSE}_{\text{unsup}}$ (Gao et al., 2021)	768	Dense	44.63	38.34	9.84	9.88	5.50	25.72	22.32
GloVe (Pennington et al., 2014)	300	Dense	44.27	36.30	10.09	13.87	8.04	29.58	23.69
$\text{SimCSE}_{\text{sup}}$ (Gao et al., 2021)	768	Dense	47.86	39.33	10.41	12.42	7.53	29.59	24.52
LLM2Vec (BehnamGhader et al., 2024)	4,096	Dense	63.48	51.73	28.56	26.29	10.39	66.36	41.14
$\text{SBERT}_{\text{new}}$ (Reimers and Gurevych, 2019)	768	Dense	88.74	47.13	37.27	32.25	21.82	62.64	48.31
$\text{ModernBERT}_{\text{emb}}$ (Warner et al., 2024)	1,024	Dense	87.39	46.56	45.18	33.99	21.40	69.98	50.75
AngIE (Li and Li, 2024)	1,024	Dense	90.43	66.15	44.84	38.65	22.98	74.07	56.19
text-embedding-ada-002 (OpenAI, 2022)	1,536	Dense	92.18	58.05	55.00	42.07	23.11	77.77	58.03
(Interpretable Embeddings)									
Bag-of-Words (with BERT vocabulary)	28,996	Sparse	29.79	34.25	3.99	21.51	6.79	47.36	23.95
BM25 (Robertson et al., 2009)	†N/A	Sparse	68.42	49.28	25.14	**32.08**	15.78	**68.70**	43.23
QAEmb-MBQA (Benara et al., 2024)	10,654	0/1 Emb.	40.51	34.75	8.23	3.87	3.74	12.01	17.19
CQG-MBQA (Sun et al., 2025)	9,614	0/1 Emb.	62.21	47.75	18.63	9.74	8.67	32.80	29.97
LDIR (with $\text{ModernBERT}_{\text{emb}}$)	200	Dense	57.65	33.62	21.06	16.05	11.27	49.59	31.54
LDIR (with $\text{ModernBERT}_{\text{emb}}$)	500	Dense	64.75	37.91	26.34	18.06	13.87	52.13	35.84
LDIR (with $\text{SBERT}_{\text{new}}$)	200	Dense	78.24	42.25	25.94	25.35	18.34	55.42	40.92
LDIR (with $\text{SBERT}_{\text{new}}$)	500	Dense	82.01	44.89	30.52	26.06	**19.74**	56.25	43.24
LDIR (with AngIE)	200	Dense	**83.11**	57.11	30.86	27.06	15.44	59.85	45.57
LDIR (with AngIE)	500	Dense	82.00	**60.10**	**33.34**	24.67	17.72	62.26	**46.68**

Table 4: Comparison between baselines and LDIR on retrieval tasks. †: BM25 is a lexical retrieval algorithm and does not generate fixed-dimensional vectors.

Model	Dim.	Type	V-Measure (Clustering)							
			TNG	SE-P2P	BR-P2P	BR-S2S	MR-P2P	MR-S2S	RD-P2P	Avg.
(Black-Box Embeddings)										
$\text{SimCSE}_{\text{unsup}}$ (Gao et al., 2021)	768	Dense	23.21	28.50	24.90	19.55	23.60	21.97	45.14	26.70
GloVe (Pennington et al., 2014)	300	Dense	25.83	28.51	29.27	19.18	26.12	20.38	35.82	26.44
$\text{BERT}_{\text{base}}$ (Devlin et al., 2019)	768	Dense	23.35	26.55	30.12	24.77	26.09	23.60	43.32	28.26
$\text{SimCSE}_{\text{sup}}$ (Gao et al., 2021)	768	Dense	34.86	29.45	30.15	24.67	26.25	24.12	47.74	31.03
LLM2Vec (BehnamGhader et al., 2024)	4,096	Dense	32.02	36.36	38.39	31.31	31.47	27.87	61.67	37.01
$\text{SBERT}_{\text{new}}$ (Reimers and Gurevych, 2019)	768	Dense	47.47	33.13	36.99	33.21	34.25	32.24	54.80	38.87
$\text{ModernBERT}_{\text{emb}}$ (Warner et al., 2024)	1,024	Dense	51.26	34.73	39.47	34.67	34.40	31.63	64.68	41.55
AngIE (Li and Li, 2024)	1,024	Dense	51.72	36.72	39.38	37.23	33.22	31.18	65.35	42.11
text-embedding-ada-002 (OpenAI, 2022)	1,536	Dense	58.14	36.88	38.03	36.53	32.70	31.27	67.96	43.07
(Interpretable Embeddings)										
Bag-of-Words (with BERT vocabulary)	28,996	Sparse	8.52	17.64	4.70	3.32	11.39	13.05	15.67	10.61
QAEmb-MBQA (Benara et al., 2024)	10,654	0/1 Emb.	36.72	25.68	24.66	21.16	25.53	22.85	46.57	29.02
CQG-MBQA (Sun et al., 2025)	9,614	0/1 Emb.	40.00	28.22	34.88	31.13	31.02	28.71	54.40	35.48
LDIR (with $\text{ModernBERT}_{\text{emb}}$)	200	Dense	29.72	37.03	25.93	21.06	31.12	27.65	52.65	32.17
LDIR (with $\text{ModernBERT}_{\text{emb}}$)	500	Dense	30.18	37.36	28.40	23.53	32.42	29.27	52.77	33.42
LDIR (with AngIE)	200	Dense	36.93	**40.60**	34.27	31.61	33.62	32.84	**58.88**	38.39
LDIR (with AngIE)	500	Dense	37.63	38.82	34.82	33.05	34.79	33.63	57.92	38.67
LDIR (with $\text{SBERT}_{\text{new}}$)	200	Dense	36.72	39.44	35.81	30.81	36.20	33.98	52.60	37.94
LDIR (with $\text{SBERT}_{\text{new}}$)	500	Dense	**40.43**	37.86	**38.07**	**33.32**	**37.63**	**35.63**	52.54	**39.36**

Table 5: Comparison between baselines and LDIR on clustering tasks.

dings baseline CQG-MBQA (29.97). This shows that, although the relatedness scores are computed through general and symmetric semantic similarity, LDIR can still encode the unsymmetrical features of relationship between the query and the document in retrieval tasks.

We also notice that the traditional sparse representation method BM25 performs well in some tasks, such as achieving nDCG@10 results of 32.08 and 68.70 on NFC and SciFact, respectively. This indicates that embeddings based on sparse representation are still useful in some specific retrieval scenarios and have better interpretability. Nevertheless, our best averaged results are still slightly better than BM25 algorithm.

Clustering. The results are shown in Table 5. Different from STS and retrieval tasks, LDIR with $\text{SBERT}_{\text{new}}$ encoder gives the best results (39.36), outperforming the CQG-MBQA baseline (35.48) by 10.9%. It also achieve comparable results to the embeddings by the backbone model $\text{SBERT}_{\text{new}}$ (38.87) and ModernBERT (41.55), validating the superior representation compactness of LDIR in unsupervised clustering. For ModernBERT and AngIE, the performance of LDIR with these encoders decrease, which may due to the trans-

Model	Dim.	Type	Perf.	Cognitive Load ↓							
				STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Bag-of-Words (with BERT vocabulary)	28,996	Sparse	56.22	8	4	6	5	8	7	6	6
QAEmb-MBQA (Benara et al., 2024)	10,654	0/1 Emb.	65.20	1,626	1,571	1,625	1,443	1,577	1,408	1,018	1,467
CQG-MBQA (Sun et al., 2025)	9,614	0/1 Emb.	77.60	481	439	458	426	478	446	413	449
CQG-MBQA (Sun et al., 2025)	†1,000	0/1 Emb.	75.24	48	44	43	41	44	42	37	43
LDIR (with AngIE, binarization)	500	0/1 Emb.	75.87	15	13	14	12	14	13	14	14
LDIR (with AngIE, binarization)	200	0/1 Emb.	71.34	6	5	5	4	5	5	5	5
LDIR (with AngIE, no binarization)	500	Dense	82.82	50	52	48	47	51	43	38	47
LDIR (with AngIE, no binarization)	200	Dense	81.99	15	16	15	14	15	13	12	14

Table 6: Evaluation on cognitive load using STS tasks. †: The corresponding performance and cognitive load with 1000 dimensional embeddings are not provided and we estimate them from the curves in Sun et al. (2025).

formation of relative representations in the low-dimensional space leads to a weakening of clustering characteristics for the high-dimensional representation space of these two embeddings.

4.3 Evaluation on Cognitive Load

We follow Sun et al. (2025) by using the "cognitive load" for measuring the interpretability. Formally, cognitive load is defined as the inner product of two binary embedding vectors μ and υ:

$$\text{cognitive load} = \langle \mu, \upsilon \rangle = \sum_{i=1}^{m} \mu_i \upsilon_i, \qquad (7)$$

where a smaller cognitive load indicates stronger model interpretability, which means we only need to focus on fewer but more representative anchors to understand the text representation. This metric is originally used for binary embeddings and we transfer LDIR to binary embeddings by setting top-k ($k = 25$ according to the average) high values as 1 and the other values in different dimensions as 0.

The results are shown in Table 6. Firstly, bag-of-words gives a very cognitive load with 6, showing the highest interpretability. CQG-MBQA exhibits a cognitive load of 449 with 9000 dimensions, which reduces to 43 with 1000 dimensions, with a sacrifice of overall performance. Our LDIR with dimensions of 500 and 200, giving cognitive load values of 14 and 5 after binarization, respectively, showing a favorable performance (71.34∼75.87) while maintaining a low cognitive load.

For LDIR without binarization, we find that it gives 47 and 14 cognitive load directly through Eq. 7, although this metric is not applicable to non-binary embeddings. Nevertheless, how to define the "interpretability" for text embeddings and design more suitable metrics that reflects interpretability for dense text embeddings still remains a valuable and open question.

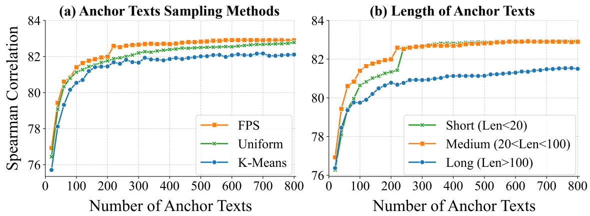

Figure 3: Comparison of different anchor texts sampling methods (a) and different settings of anchor texts length (b). The horizontal axis represents the different number (20∼800) of different anchor texts.

5 Analyses and Discussions

In our main experiments, we remove the long texts in corpus and use FPS to extract the anchor texts. We analysis the impact of sampling methods (Section 5.1) and the length of anchor texts (Section 5.2) below. Then we discuss how to obtain a more interpretable dense embeddings with the calculation of fine-grained relatedness (Section 5.3), compare different metrics on relatedness (Section 5.4). Finally, we show a case study (Section 5.5) and discuss the usage of relative representation based embeddings (Section 5.6).

5.1 Anchor Texts Sampling Methods

We compare three anchor texts sampling methods: FPS, uniform sampling, and K-Means . The results are shown in Figure 3(a). The trends of the three methods across different numbers of anchors were similar in general. FPS generally performed the best, with its performance converging with uniform sampling at 800 dimensions, while K-Means performed relatively poorly. The advantage of FPS in the low-dimensional range stems from its maximum diversity sampling strategy. The convergence of uniform sampling with FPS at 800 dimensions may be related to the properties of high-dimensional spaces when the dimensionality is sufficiently high, random uniform sampling can

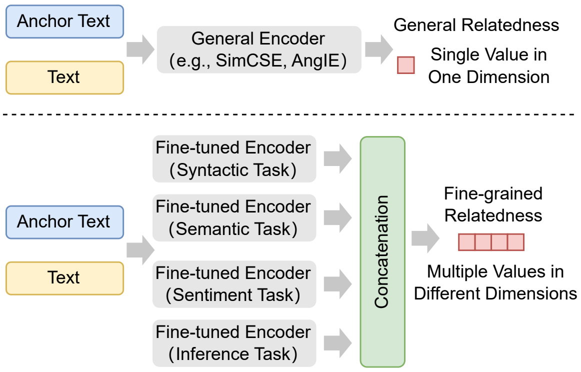

Figure 4: Comparison between the calculation of general (top) and fine-grained (bottom) relatedness scores.

Model	STS12	STS13	STS14	Avg.
Bag-of-Words (with BERT vocabulary)	44.75	52.06	54.78	50.53
QAEmb-MBQA (Benara et al., 2024)	59.40	63.19	57.68	60.09
CQG-MBQA (Sun et al., 2025)	69.21	80.19	73.91	74.44
LDIR (with fine-grained relatedness)	57.88	67.33	58.96	61.39

Table 7: The results of LDIR using the concatenated 800 dimensional embeddings with fine-grained relatedness.

approximately cover the boundary regions of the semantic space. Notably, the K-Means method, constrained by the high similarity of cluster centers, resulted in insufficient semantic diversity in the anchor set compared to the other two methods, leading to suboptimal performance.

5.2 The Impact of Length of Anchor Texts

We investigate the impact of different length of anchor texts, including short (less than 20 tokens), medium (20∼100 tokens), and long (longer than 100 tokens) texts. The results are shown in Figure 3(b). We observe that for different settings of length, the performance improved rapidly using less than 200 anchor texts, after which the improvement is largely limited. The performance gap between short texts and medium texts narrowed and almost converged with 240 or more anchor texts, with both significantly outperforming long texts. We hypothesize that this may be attributable to the more intricate semantics and the potential inclusion of greater noise within longer texts in the corpus.

5.3 Fine-grained Relatedness Calculation

In Eq. 5, the relatedness score is rather general, which reflects the overall semantic correlation between anchor text a_j and text t. We further propose calculate more fine-grained relatedness score by applying different encoders tuned on different tasks, as shown in Figure 4 (bottom).

Metrics/Distance	Type	STS Results in Avg.
Edit Distance	surface-based	17.99
Jaccard	surface-based	22.34
Sokalsneath	vector-based & binary	79.04
Jaccard	vector-based & binary	79.49
Hamming	vector-based & binary	80.00
Dice	vector-based & binary	80.07
Chebyshev	vector-based & dense	79.33
Euclidean	vector-based & dense	80.76
Manhattan	vector-based & dense	81.19
Cosine (this work)	vector-based & dense	82.82

Table 8: The results of LDIR using different surface-based and vector-based metrics.

We use tasks in the general language understanding evaluation (GLUE; Wang et al., 2019) benchmark, which tests the text understanding ability from different perspectives. In particular, we select CoLA, STS-B, SST-2, and QNLI subtasks such that the fine-tuned models can encode syntactic, semantic, sentiment, and natural language inference features in fine-grained levels. Different models are fine-tuned on the training data and serve as encoders for calculating relatedness. The relatedness scores via 200 anchor texts are concatenated as a final 800-dimensional dense embeddings.

The results on STS12, STS13, and STS14 are shown in Table 7. We find that LDIR with fine-grained relatedness still outperforms the QAEmb-MBQA baseline (61.39 vs. 60.09). However, its overall performance decreases in STS tasks, underperforming the CQG-MBQA baseline (61.39 vs. 74.44). This show that, although the fine-grained relatedness scores have improved interpretability, their overall performance will not be enhanced in general semantic similarity tasks, which also reflects the trade-off between performance and interpretability, as discussed in Table 2.

5.4 Surface-based and Vector-based Metrics

We further try surface-based metrics as well as other various vector-based metrics for calculating the relatedness, and the results for 500 dimensional embeddings are shown in Table 8. We find that surface-based distances cannot effectively represent semantic relationships, where the results are very poor. On the other hand, other vector-based distance (even when applied to LDIR after binarization) consistently show close results, despite differences in distances calculation against the cosine distances in downstream tasks.

Anchor Text #1:	A novel, high-performing architecture for end-to-end named entity recognition and relation extraction that is fast to train.
Anchor Text #2:	Predicting affective states expressed through an emote-aloud procedure from autotutor's mixed-initiative dialogue.
Anchor Text #3:	Planar High-Gain Dielectric-Loaded Antipodal Linearly Tapered Slot Antenna for E- and W-Band Gigabyte Point-to-Point Wireless Services
Anchor Text #4:	Design of a Monopulse Antenna Using a Dual V-Type Linearly Tapered Slot Antenna (DVLTSA).

Texts (↓) and LDIR Embeddings (→)		**Dim #1**	**Dim #2**	**Dim #3**	**Dim #4**
Text A:	Morphological Embeddings for Named Entity Recognition in Morphologically Rich Languages.	0.7554	0.5312	0.3801	0.3607
Text B:	Design Approach to a Novel Dual-Mode Wideband Circular Sector Patch Antenna.	0.3941	0.3373	0.6874	0.6561

Table 9: LDIR embeddings of text A and text B. We only show values on four dimensions for brevity.

5.5 Case Study

Table 9 shows a case of four-dimensional LDIR embeddings of two title texts from SCIDocs. Text A has the highest relevance to anchor text 1, while text B is more related to anchor texts 3 and 4, resulting in the difference in their LDIR embeddings.

5.6 The Usage of Relative Representation

Our approach is based on transforming relative representations using an existing backbone model, where we consider its potential applications here briefly. For instance, some studies have shown that it is possible to recover original training data or sensitive information from embeddings themselves (Morris et al., 2023; Li et al., 2023; Huang et al., 2024). Therefore, directly exposing embedding interfaces may lead to data privacy or leakage issues. Consider a scenario where an embedding provider (such as text-embedding-Ada series by OpenAI) does not want directly provide users with high-dimensional raw embeddings outputs. Instead, they could offer low-dimensional relative representation vectors based on anchor text (which can be provided by the provider or defined by the user). This approach 1) can help avoid some data privacy or leakage concerns; 2) performs well in certain downstream applications; 3) allows for obtaining more text representations with the same storage and computational cost with low-dimensionality; 4) provides interpretability and reliability with the flexibility of anchor text.

6 Conclusion

We propose LDIR, a low-dimensional dense and interpretable text embeddings with relative representations. The numerical values in LDIR represent the correlation between the text and the anchor texts automatically obtained through farthest point sampling, which improves the expressiveness at a low cost compared with QA-based 0/1 embeddings. Across multiple tasks and datasets, LDIR demonstrates better semantic expressiveness compared to multiple interpretable embedding baselines. We also discuss the need for more suitable interpretability metrics and how to further enhance the interpretability of LDIR, which can serve as our future work for interpretable text embeddings.

Limitations

Interpretable text embeddings is a relatively new research topic and the baselines of interpretable text embeddings we used is limited. Also, as we mentioned, there is still room for improvement in the interpretability of our embeddings. According to our approach, the anchor texts are sampled from the entire corpus, and we can further optimize these anchor texts for different downstream tasks for better task-related expressiveness. Additionally, the human evaluation and application scenarios of interpretable text embeddings can be further explored, which is also crucial for explainable AI.

Acknowledgments

This work was supported in parts by NSFC (62306161, U21B2023), Shenzhen Science and Technology Program (KJZD20240903100022028, KQTD20210811090044003, RCJC20200714114435012), and Scientific Development Funds from Shenzhen University.

References

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

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

Eneko Agirre, 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-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: 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 385–393, Montréal, Canada. Association for Computational Linguistics.

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

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. 2021. Revisiting model stitching to compare neural representations. *Advances in neural information processing systems*, 34:225–236.

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*.

Vinamra Benara, Chandan Singh, John Xavier Morris, Richard Antonello, Ion Stoica, Alexander Huth, and Jianfeng Gao. 2024. Crafting interpretable embeddings for language neuroscience by asking LLMs questions. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

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*, pages 716–722, Cham. Springer International Publishing.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

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

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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 55–65, Hong Kong, China. Association for Computational Linguistics.

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

Yu-Hsiang Huang, Yuche Tsai, Hsiang Hsiao, Hong-Yi Lin, and Shou-De Lin. 2024. Transferable embedding inversion attack: Uncovering privacy risks in text embeddings without model queries. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4193–4205, Bangkok, Thailand. Association for Computational Linguistics.

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.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. NV-embed: Improved techniques for training LLMs as generalist embedding models. In *The Thirteenth International Conference on Learning Representations*.

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 (Volume 1: Long Papers)*, pages 10141–10157, Bangkok, Thailand. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Haoran Li, Mingshi Xu, and Yangqiu Song. 2023. Sentence embedding leaks more information than you expect: Generative embedding inversion attack to recover the whole sentence. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 14022–14040, Toronto, Canada. Association for Computational Linguistics.

Xianming Li and Jing Li. 2024. AoE: Angle-optimized embeddings for semantic textual similarity. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1825–1839, Bangkok, Thailand. Association for Computational Linguistics.

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.

Mayug Maniparambil, Raiymbek Akshulakov, Yasser Abdelaziz Dahou Djilali, Mohamed El Amine Seddik, Sanath Narayan, Karttikeya Mangalam, and Noel E. O'Connor. 2024. Do vision and language encoders represent the world similarly? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14334–14343.

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'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Tomas Mikolov. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 3781.

John Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander Rush. 2023. Text embeddings reveal (almost) as much as text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12448–12460, Singapore. Association for Computational Linguistics.

Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. 2023. Relative representations enable zero-shot latent space communication. In *The Eleventh International Conference on Learning Representations*.

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.

Antonio Norelli, Marco Fumero, Valentino Maiorca, Luca Moschella, Emanuele Rodola, and Francesco Locatello. 2023. Asif: Coupled data turns unimodal models to multimodal without training. *Advances in Neural Information Processing Systems*, 36:15303–15319.

OpenAI. 2022. Openai api embeddings.

Ajay Patel, Delip Rao, Ansh Kothary, Kathleen McKeown, and Chris Callison-Burch. 2023. Learning interpretable style embeddings via prompting LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15270–15290, Singapore. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. 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.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Yiqun Sun, Qiang Huang, Yixuan Tang, Anthony Kum Hoe Tung, and Jun Yu. 2025. A general framework for producing interpretable semantic text embeddings. In *The Thirteenth International Conference on Learning Representations*.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663*.

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 (Volume 1: Long Papers)*, pages 12135–12148, Toronto, Canada. Association for Computational Linguistics.

A Checkpoints of Baseline Models

In Table 10, we list the checkpoints of baseline models we used in experiments.

Model	Checkpoint
BERT	https://huggingface.co/google-bert/bert-base-uncased
GloVe	https://huggingface.co/sentence-transformers/average_word_embeddings_glove.6B.300d
SimCSE$_{\text{unsup}}$	https://huggingface.co/princeton-nlp/sup-simcse-bert-base-uncased
SimCSE$_{\text{sup}}$	https://huggingface.co/princeton-nlp/unsup-simcse-bert-base-uncased
WhitenedCSE	https://huggingface.co/SupstarZh/whitenedcse-bert-base
SBERT$_{\text{ori}}$	https://huggingface.co/sentence-transformers/all-mpnet-base-v1
SBERT$_{\text{new}}$	https://huggingface.co/sentence-transformers/all-mpnet-base-v2
ModernBERT$_{\text{emb}}$	https://huggingface.co/lightonai/modernbert-embed-large
text-embedding-ada-002	https://openai.com/index/new-and-improved-embedding-model/
AngIE	https://huggingface.co/WhereIsAI/UAE-Large-V1
LLM2Vec	https://huggingface.co/McGill-NLP/LLM2Vec-Meta-Llama-3-8B-Instruct-mntp-unsup-simcse
BM25	https://github.com/xhluca/bm25s

Table 10: Checkpoints of baseline models we used in experiments.