Mitigating Language Bias in Multilingual Sentence Embeddings for Cross-Lingual Similarity Estimation

Kanade Nonomura[†] **Keita Fukushima**[†] **Risa Kondo**[†] **Tomoyuki Kajiwara**[†‡]
[†] Ehime University, Japan [‡] D3 Center, The University of Osaka, Japan
{nonomura@ai.cs., fukushima@ai.cs., kondo@ai.cs., kajiwara@cs.}ehime-u.ac.jp

Abstract

We disentangle multilingual sentence embeddings into language-dependent and language-agnostic components, leveraging the latter to improve cross-lingual similarity estimation. Previous studies on this approach have trained disentanglers by combining intra-component constraints, which either align or disalign language-dependent embeddings or language-agnostic embeddings, with inter-component constraints across both embeddings. However, when and how these constraints are effective remains unclear. Our experiments on sentence similarity estimation and machine translation quality estimation revealed that while intra-component constraints and the combination of both constraints are effective for encoder-based multilingual sentence embeddings, inter-component constraints are effective for decoder-based ones. Furthermore, our detailed analysis revealed distinct roles: intra-component constraints improve uniformity within the embedding space, while inter-component constraints enhance cross-lingual alignment between parallel sentences.

1 Introduction

Multilingual sentence embeddings (Feng et al., 2022; Wang et al., 2024; Lee et al., 2025; Zhang et al., 2025) have been widely adopted in a variety of natural language processing tasks, including machine translation quality estimation (Specia et al., 2020) and cross-lingual semantic textual similarity estimation (Cer et al., 2017). However, such embeddings inevitably encode language-dependent representations, leading them to cluster by language (Tiyajamorn et al., 2021) and degrading performance on cross-lingual tasks. To address this issue, previous studies (Tiyajamorn et al., 2021; Kuroda et al., 2022; Ki et al., 2024; Fukushima et al., 2025; Nonomura et al., 2026) have trained disentanglers using bilingual corpora, as shown in Figure 1, aiming to acquire language-agnostic embeddings

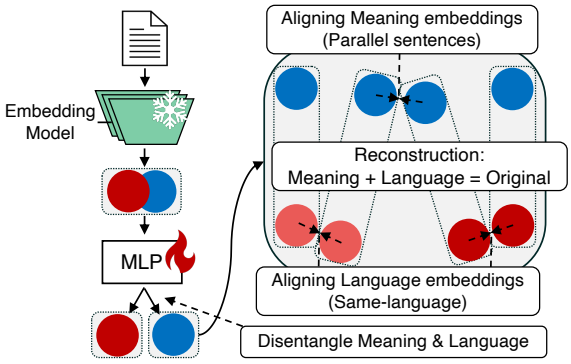

Figure 1: Overview of disentangling meaning and language embedding from multilingual embedding.

(hereafter referred to as meaning embeddings) independent of language-dependent embeddings (hereafter referred to as language embeddings).

Among this line of work, we focus on SEED (Fukushima et al., 2025), which achieves state-of-the-art performance in embedding disentanglement. The loss functions used in SEED can be broadly categorized into two distinct types, each sufficient for disentanglement: intra-component constraints and inter-component constraints. Intra-component constraints operate within a single component, defined specifically for either meaning embeddings or language embeddings. Concretely, they maximize the similarity between meaning embeddings of parallel sentences, while simultaneously maximizing the similarity between language embeddings of sentences written in the same language. Inter-component constraints, in contrast, are defined across meaning embeddings and language embeddings. Under the assumption that a sentence embedding can be decomposed into the sum of a meaning embedding and a language embedding, these constraints operate by swapping meaning embeddings between parallel sentences or swapping language embeddings between sentences of the same language. This design encourages similarity between

*Proceedings of the 15th Joint Conference on Lexical and Computational Semantics (*SEM 2026)*, pages 385–394

embeddings that are related through such swapping operations. Importantly, both types of constraints are individually sufficient to disentangle meaning embeddings and language embeddings. However, it remains unclear how the impact of each constraint differs between encoder-based embedding models and decoder-based ones, and what geometric properties (Wang and Isola, 2020) in the embedding space account for these differences.

In this study, we investigate these questions via experiments on machine translation quality estimation (Specia et al., 2020) and cross-lingual semantic textual similarity estimation (Cer et al., 2017) tasks. We further analyze the effects of each constraint from the perspective of embedding space geometry (Wang and Isola, 2020), and examine how their effectiveness changes when the embedding dimensionality of LLM-based models is reduced (Kusupati et al., 2022).

Experimental results show that for encoder-based models (Feng et al., 2022; Wang et al., 2024), combining both constraints is the most effective, followed by using intra-component constraints alone. In contrast, for decoder-based models (Lee et al., 2025; Zhang et al., 2025), inter-component constraints are shown to be the most effective. Furthermore, analyses of the geometric properties of the embedding space suggest that improving embedding uniformity is crucial for encoder-based models, whereas enhancing the alignment of parallel sentences is more important for decoder-based models. In addition, we analyze how downstream task performance changes with reduced embedding dimensionality for LLM-based models. The results show that inter-component constraints remain consistently the most effective across different dimensionalities, confirming that the differences in effective constraints for embedding disentanglement are not attributable to embedding dimensionality.

2 Related Work

2.1 Multilingual Text Embedding Models

Pre-trained models such as mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) are known to suffer from anisotropy, where sentence embeddings concentrate in a narrow region of the vector space (Ethayarajh, 2019; Gao et al., 2021). To address this, contrastive learning has been widely adopted to improve embedding quality (Feng et al., 2022; Wang et al., 2024). Similarly, recent LLM-based models utilizing contrastive learning have achieved state-of-the-art performance on benchmarks like Massive Multilingual Text Embedding Benchmark (Enevoldsen et al., 2025), despite originally focusing on generation (Zhang et al., 2025; Lee et al., 2025). Notably, encoder-based models typically use around 1,000 dimensions, whereas LLM-based models employ several thousand. To investigate whether the significant dimensionality gap between these architectures influences constraint effectiveness, we focus on models trained with Matryoshka Representation Learning (MRL) (Kusupati et al., 2022), which enables flexible dimensionality reduction.

2.2 Language-Agnostic Representation from Multilingual Sentence Encoders

Ideally, multilingual sentence embeddings should reflect pure language-agnostic representations that is independent of language, such that sentences with equivalent meanings are close in the embedding space. In practice, however, they often exhibit language-specific clustering (Tiyajamorn et al., 2021), which can degrade cross-lingual task performance. To address this, prior studies assume that embeddings contain both meaning and language components, and have proposed methods to disentangle them into independent factors (Tiyajamorn et al., 2021; Kuroda et al., 2022; Ki et al., 2024; Fukushima et al., 2025). DREAM (Tiyajamorn et al., 2021) utilizes two MLPs to extract meaning and language embeddings by maximizing the similarity of the former between parallel sentences and the latter within the same language, while ensuring their sum reconstructs the original embedding. MEAT (Kuroda et al., 2022) builds upon DREAM by introducing adversarial training to prevent language identification from meaning embeddings, thereby enforcing stricter removal of language-specific information. ORACLE (Ki et al., 2024) introduces an orthogonality-constrained training objective that explicitly separates semantic and language embeddings to reduce semantic leakage and improve cross-lingual semantic alignment. SEED (Fukushima et al., 2025) points out that using separate MLPs to obtain meaning embeddings and language embeddings, as in previous methods, can result in missing information before and after disentanglement. To address this issue, SEED defines language embeddings as the difference between the original sentence embeddings and the meaning embeddings, thereby preventing missing information across the disen-

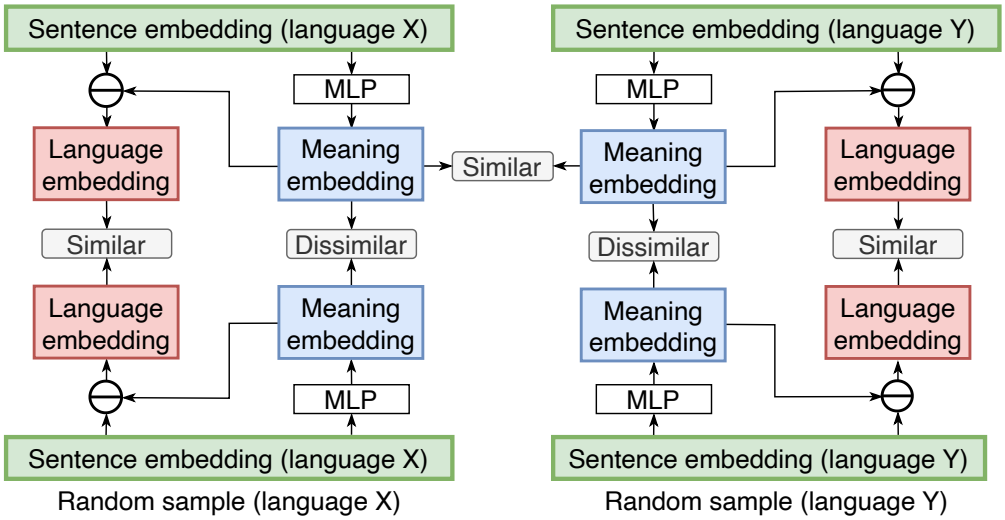

Figure 2: Overview of the intra-component constraints. The two sentence embeddings shown at the top of the figure correspond to a parallel sentence pair sampled from the parallel corpus. If the sentence on the left corresponds to language X, a sentence embedding randomly selected from the set of sentence embeddings in language X is provided at the bottom left. The same procedure is applied symmetrically to the other side.

tanglement process. Furthermore, a recent study has comprehensively analyzed the effectiveness of these disentanglement methods by systematically categorizing aspects such as model architectures (e.g., encoders and decoders) and the presence of fine-tuning like contrastive learning (Nonomura et al., 2026). Another line of work leverages disentanglement frameworks to develop domain-specific cross-lingual sentence encoders from existing multilingual and domain-specific monolingual encoders, avoiding costly multilingual in-domain pre-training (Kondo et al., 2025). While these methods typically employ multi-task learning, the distinct roles of individual loss functions and their compatibility with different embedding model architectures remain unclear. In this work, we address this issue by decomposing the loss functions used in SEED into intra-component constraints and inter-component constraints, and systematically analyzing their effects.

3 Loss Functions for Disentangling Multilingual Sentence Embeddings

The loss functions used to train SEED (Fukushima et al., 2025) can be broadly categorized into two types: intra-component constraints and inter-component constraints. Intra-component constraints are defined within either meaning embeddings or language embeddings, whereas inter-component constraints are defined across both components. Both types of constraints are designed such that meaning embeddings become similar between parallel sentences, and language embeddings become similar between sentences written in the same language. In this section, we first describe the model architecture following the SEED framework. We then explain the intra-component constraints and inter-component constraints that are analyzed in this study.

3.1 Model Architecture

Let x be a sentence in language X and y be a sentence in language Y. Each sentence embedding contains both language-agnostic information and language-specific information. To disentangle these as independent components, we assume that a sentence embedding is composed as the sum of a meaning embedding $\boldsymbol{e}^{(m)}$ and a language embedding $\boldsymbol{e}^{(l)}$. Following SEED (Fukushima et al., 2025), we use a extractor for meaning embeddings implemented as a multi-layer perceptron (MLP) to obtain the meaning embedding. The language embedding is then defined as the residual between the original sentence embedding and the meaning embedding:

$$\boldsymbol{e}^{(m)} = \mathrm{MLP}(\boldsymbol{e}), \quad \boldsymbol{e}^{(l)} = \boldsymbol{e} - \boldsymbol{e}^{(m)}. \tag{1}$$

Based on this formulation, we define loss functions that encourage meaning embeddings to capture language-agnostic information and language embeddings to capture language-specific information.

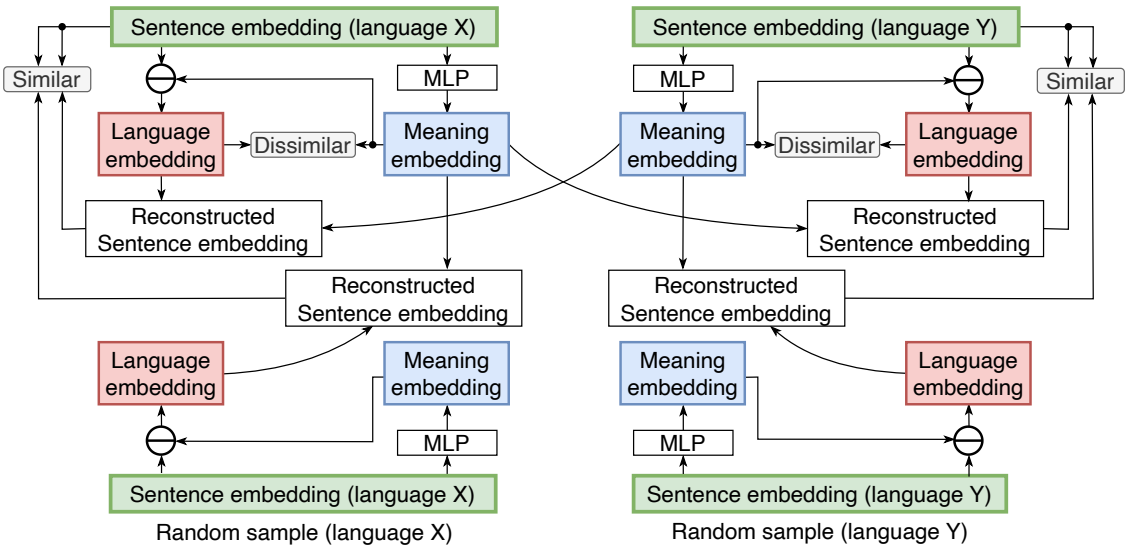

Figure 3: Overview of the inter-component constraints. The two sentence embeddings shown at the top of the figure correspond to a parallel sentence pair sampled from the parallel corpus. If the sentence on the left corresponds to language X, a sentence embedding randomly selected from the set of sentence embeddings in language X is provided at the bottom left, and the same applies symmetrically to the other side.

3.2 Intra-Component Constraints

An overview of the intra-component constraints is illustrated in Figure 2. The intra-component constraint L_{intra} is defined within either meaning embeddings or language embeddings, and is formulated as:

$$L_{\text{intra}} = L_{\text{mean}} + L_{\text{lang}}. \tag{2}$$

Let (x_i, y_i) denote the i-th pair of parallel sentences in a batch. Let j denote an index different from i within the same batch.

Meaning Embedding Loss For meaning embeddings, parallel sentences (x_i, y_i) share the same meaning; therefore, their similarity should be maximized. In contrast, different sentences within the same language, (x_i, x_j) and (y_i, y_j), generally convey different meanings, and their similarity should be minimized. Accordingly, we define:

$$\begin{aligned} L_{\text{mean}} = 2 &\left(1 - \cos\left(\boldsymbol{e}_{x_i}^{(m)}, \boldsymbol{e}_{y_i}^{(m)}\right)\right) \\ &+ \max\left(0, \cos\left(\boldsymbol{e}_{x_i}^{(m)}, \boldsymbol{e}_{x_j}^{(m)}\right)\right) \\ &+ \max\left(0, \cos\left(\boldsymbol{e}_{y_i}^{(m)}, \boldsymbol{e}_{y_j}^{(m)}\right)\right). \end{aligned} \tag{3}$$

Language Embedding Loss For language embeddings, sentences written in the same language, (x_i, x_j) and (y_i, y_j), share language-specific information. Therefore, their similarity should be maximized. We define:

$$L_{\text{lang}} = 2 - \cos\left(\boldsymbol{e}_{x_i}^{(l)}, \boldsymbol{e}_{x_j}^{(l)}\right) - \cos\left(\boldsymbol{e}_{y_i}^{(l)}, \boldsymbol{e}_{y_j}^{(l)}\right). \tag{4}$$

3.3 Inter-Component Constraints

An overview of the inter-component constraints is shown in Figure 3. The inter-component constraint L_{inter} is defined across both meaning embeddings and language embeddings:

$$L_{\text{inter}} = L_{\text{cross}} + L_{\text{sep}}. \tag{5}$$

Cross-Reconstruction Loss Parallel sentences (x_i, y_i) share identical meaning despite being written in different languages. Therefore, combining the language embedding of x_i with the meaning embedding of y_i (or vice versa) should reconstruct the original sentence embedding $\boldsymbol{e}_{x_i}$ (or $\boldsymbol{e}_{y_i}$), and similarly for swaps between same-language sentences. On the other hand, sentences in the same language, (x_i, x_j) and (y_i, y_j), are assumed to share language-specific information but differ in meaning. Thus, combining the meaning embedding of x_i with the language embedding of x_j should reconstruct $\boldsymbol{e}_{x_i}$, and similarly for y_i. We define the cross-reconstruction loss as:

$$\begin{aligned} L_{\text{cross}} = 4 &- \cos\left(\boldsymbol{e}_{x_i}, \boldsymbol{e}_{y_i}^{(m)} + \boldsymbol{e}_{x_i}^{(l)}\right) \\ &- \cos\left(\boldsymbol{e}_{y_i}, \boldsymbol{e}_{x_i}^{(m)} + \boldsymbol{e}_{y_i}^{(l)}\right) \\ &- \cos\left(\boldsymbol{e}_{x_i}, \boldsymbol{e}_{x_i}^{(m)} + \boldsymbol{e}_{x_j}^{(l)}\right) \\ &- \cos\left(\boldsymbol{e}_{y_i}, \boldsymbol{e}_{y_i}^{(m)} + \boldsymbol{e}_{y_j}^{(l)}\right). \end{aligned} \tag{6}$$

Separation Loss To prevent the extracted meaning embeddings and language embeddings from

encoding overlapping information, we minimize their similarity. We define the separation loss as:

$$L_{\text{sep}} = \max\left(0, \cos\left(\boldsymbol{e}_{x_i}^{(m)}, \boldsymbol{e}_{x_i}^{(l)}\right)\right) + \max\left(0, \cos\left(\boldsymbol{e}_{y_i}^{(m)}, \boldsymbol{e}_{y_i}^{(l)}\right)\right). \quad (7)$$

4 Analysis of Task Performance

In this section, we investigate the effects of Intra-component constraints, Inter-component constraints, and their combination on task performance for encoder-based models and LLM-based models. We evaluate the models on the WMT20 QE task (Specia et al., 2020) and the SemEval-2017 STS task (Cer et al., 2017). QE is a task that estimates translation quality without using reference translations. Since our method is trained solely on parallel corpora, we address the task in an unsupervised[1] setting. STS is a task that estimates the similarity in meaning between two given sentences. For both tasks, cosine similarity is computed based on the meaning representations of sentence pairs, and model performance is evaluated using Pearson correlation with human evaluation scores.

4.1 Experimental Setup

Dataset WMT20 QE task includes six[2] language pairs. For each language pair, a dataset consisting of 1,000 source–target sentence pairs with human-assigned quality scores is provided. The machine translation systems evaluated in this task are Transformer models (Vaswani et al., 2017) trained using the fairseq toolkit (Ott et al., 2019). For training our models, we used a subset of the parallel corpora available for the task.[3] The data statistics are shown in Table 3.

SemEval-2017 STS task includes seven[4] language pairs. For each language pair, 250 sentence pairs with human evaluation scores are provided for evaluation. For model training, we extract parallel data from the Tatoeba corpus,[5] following prior work (Tiyajamorn et al., 2021; Fukushima et al., 2025). The data statistics are shown in Table 3.

Models In our experiments, we freeze the parameters of the backbone models and only train the MLP, which consists of a single linear layer. For sentence embeddings, we use LaBSE (Feng et al., 2022) and mE5-large (Wang et al., 2024) as encoder-based models, and Gemini Embedding (Lee et al., 2025) and Qwen3-Embedding-8B (Zhang et al., 2025) as LLM-based models. For encoder-based models, the input sentence is directly fed into the model. We use the final-layer CLS token for LaBSE and mean pooling over the final-layer token representations for mE5-large. For Gemini Embedding, we use the API and specify the task type as STS when obtaining embeddings. For Qwen3-Embedding-8B, following the official evaluation repository,[6] we prepend the instruction `Retrieve semantically similar text` to the input sentence and use the final-layer EOS token as the sentence embedding. Both LLM-based models support multiple embedding dimensionalities via MRL (Kusupati et al., 2022); in this study, we use the maximum dimensionalities of 3,072 and 4,096, respectively.

Compared Methods We compare training with Intra-component constraints, Inter-component constraints, and their combination. In addition, we use a baseline in which the original sentence embeddings are directly used without separating meaning representations. The combination of both constraints follows the same configuration as SEED (Fukushima et al., 2025).

Hyperparameters We use HuggingFace Transformers (Wolf et al., 2020) for training. The batch size is set to 512, the optimizer is Adam (Kingma and Ba, 2015), and the learning rate is $1e^{-4}$. Ten percent of the training data is randomly sampled as validation data. Training is terminated if the validation loss does not improve for three consecutive epochs.

4.2 Results

WMT20 QE Task Table 1 presents the experimental results. Overall, all constraint-based meth-

[1] In the supervised setting, models are trained on triplets of source sentence, target sentence, and human evaluation score, whereas in the unsupervised setting, human evaluation scores are not used.

[2] They consist of high-resource pairs English–German (en-de) and English–Chinese (en-zh), medium-resource pairs Romanian–English (ro-en) and Estonian–English (et-en), and low-resource pairs Nepali–English (ne-en) and Sinhala–English (si-en). https://github.com/facebookresearch/mlqe

[3] Randomly sampled from http://www.statmt.org/wmt20/quality-estimation-task.html in amounts comparable to prior work (Kuroda et al., 2022; Fukushima et al., 2025).

[4] They consist of high-resource pairs English–Italian (en-it) and English–Turkish (en-tr), medium-resource pairs English–German (en-de), English–Spanish (en-es), and English–French (en-fr), and low-resource pairs English–Arabic (en-ar) and English–Dutch (en-nl). https://public.ukp.informatik.tu-darmstadt.de/reimers/sentence-transformers/datasets/

[5] https://tatoeba.org/ja/

[6] https://github.com/QwenLM/Qwen3-Embedding

Model	Method	High Resource		Medium Resource		Low Resource		
		en-de	en-zh	ro-en	et-en	ne-en	si-en	Avg.
LaBSE	Baseline	0.084	0.036	0.705	0.550	0.547	0.455	0.396
	Intra	0.187	**0.193**	**0.726**	**0.584**	**0.634**	**0.564**	0.481
	Inter	0.143	0.044	0.698	0.567	0.517	0.434	0.401
	Both	**0.191**	0.192	0.725	**0.584**	0.633	**0.564**	**0.482**
mE5-large	Baseline	0.020	0.100	0.734	0.556	0.538	0.493	0.407
	Intra	**0.179**	**0.255**	0.776	0.634	0.590	**0.543**	**0.496**
	Inter	0.146	0.174	**0.787**	0.631	0.570	0.509	0.470
	Both	0.176	0.248	0.781	**0.635**	**0.591**	**0.543**	**0.496**
Gemini Embedding	Baseline	0.159	0.279	0.789	0.605	0.699	0.561	0.515
	Intra	**0.219**	0.295	0.788	0.626	0.681	0.562	0.529
	Inter	0.204	0.302	**0.798**	**0.639**	**0.701**	0.569	**0.536**
	Both	0.218	**0.310**	0.792	0.628	0.686	**0.570**	0.534
Qwen3-Embedding-8B	Baseline	0.186	0.261	0.755	0.476	0.626	0.431	0.456
	Intra	**0.222**	0.287	0.754	0.516	0.616	0.444	0.473
	Inter	0.218	0.292	**0.763**	**0.528**	**0.632**	0.451	**0.481**
	Both	**0.222**	**0.295**	0.757	0.517	0.625	**0.454**	0.478

Table 1: Experimental results on the WMT20 QE task (Pearson's correlation coefficient). "Intra" and "Inter" denote intra-component and inter-component constraints, respectively, and "Both" indicates their combination.

Model	Size	Dim.
LaBSE	471M	768
mE5-large	560M	1024
Gemini Embedding	-	3072
Qwen3-Embedding-8B	8B	4096

Table 2: Model configurations of the sentence embedding models

WMT20 QE		SemEval-2017 STS	
Language Pair	# Pairs	Language Pair	# Pairs
en-de, en-zh	1,000 k	en-it, en-tr	500 k
ro-en, et-en	200 k	en-de, en-es, en-fr	200 k
ne-en, si-en	50 k	en-ar, en-nl	30 k

Table 3: Number of sentence pairs used for training (10% reserved for validation)

ods improve the average performance compared to the no-training setting, indicating that embedding separation functions effectively. However, clear differences are observed depending on the model architecture. For encoder-based models (LaBSE and mE5-large), the combination of both constraints achieves the best performance on most language pairs, followed by the Intra-component constraint. In contrast, for LLM-based models (Gemini Embedding and Qwen3-Embedding-8B), the Inter-component constraint alone yields the highest performance for many language pairs as well as in terms of the average score. For LLM-based models, combining both constraints does not necessarily produce the best results, and adding the Intra-component constraint does not consistently lead to further improvements.

SemEval-2017 STS Task Table 4 presents the results. A similar trend to the QE task is observed in the STS task. For the encoder-based model LaBSE, the combination of both constraints achieves the highest performance across all language pairs. In contrast, for the LLM-based Gemini Embedding, the Inter-component constraint alone achieves the best performance on six language pairs as well as in the average score.

5 Analysis of Geometric Properties

Previous results suggest that differences in the geometric properties of the embedding space drive the diverging performance gains observed between encoder-based and LLM-based architectures. In this section, we analyze the differences induced by each constraint from the perspective of geometric properties of the embedding space, and investigate

Model	Method	High Resource		Medium Resource			Low Resource		
		en-it	en-tr	en-de	en-es	en-fr	en-ar	en-nl	Avg.
LaBSE	Baseline	0.760	0.748	0.721	0.692	0.759	0.705	0.755	0.734
	Intra	0.778	0.756	0.745	0.691	0.782	0.732	**0.776**	0.751
	Inter	0.750	0.745	0.710	0.686	0.749	0.695	0.753	0.727
	Both	**0.780**	**0.760**	**0.746**	**0.693**	**0.783**	**0.735**	**0.776**	**0.753**
Gemini Embedding	Baseline	0.908	0.860	0.899	0.881	0.901	0.856	0.899	0.886
	Intra	0.903	0.860	0.894	0.882	0.895	**0.871**	0.896	0.886
	Inter	**0.913**	**0.869**	**0.902**	**0.890**	**0.906**	0.863	**0.905**	**0.893**
	Both	0.908	0.864	0.899	0.884	0.901	**0.871**	0.900	0.890

Table 4: Experimental results on the SemEval-2017 STS task (Pearson's correlation coefficient). "Intra" and "Inter" denote intra-component and inter-component constraints, respectively, and "Both" indicates their combination.

what factors are effective for each architecture. As evaluation metrics, we adopt *Alignment* and *Uniformity* (Wang and Isola, 2020), which are widely used to assess representation learning methods, including contrastive learning, from a geometric viewpoint (Gao et al., 2021; Chuang et al., 2022; Li et al., 2024). Let $\tilde{h}(x)$ denote the L2-normalized embedding of the meaning embedding of an input sentence x.

5.1 Alignment and Uniformity

Alignment Alignment measures how closely parallel sentence representations are positioned in the normalized embedding space. Let x be a sentence in language X, y be its parallel sentence in language Y, and let p_{pos} denote the distribution over parallel pairs (x, y). Alignment is defined as:

$$L_{\text{align}} = \mathop{\mathbb{E}}_{(x,y)\sim p_{\text{pos}}} \left\| \tilde{h}(x) - \tilde{h}(y) \right\|^2. \quad (8)$$

A smaller value indicates that parallel sentences are closer in the embedding space, reflecting stronger cross-lingual consistency.

Uniformity Uniformity measures how evenly normalized embeddings are distributed over the hypersphere. Let p_{data} denote the marginal distribution over all sentences across language pairs. For independently sampled sentences $x, z \sim p_{\text{data}}$, Uniformity is defined as:

$$L_{\text{uniform}} = \log \mathop{\mathbb{E}}_{x,z \overset{\text{i.i.d.}}{\sim} p_{\text{data}}} e^{-2\left\| \tilde{h}(x) - \tilde{h}(z) \right\|^2}. \quad (9)$$

A smaller value indicates that embeddings are more uniformly dispersed across the hypersphere. This implies that the embedding space avoids concentration in limited regions and utilizes the space more effectively.

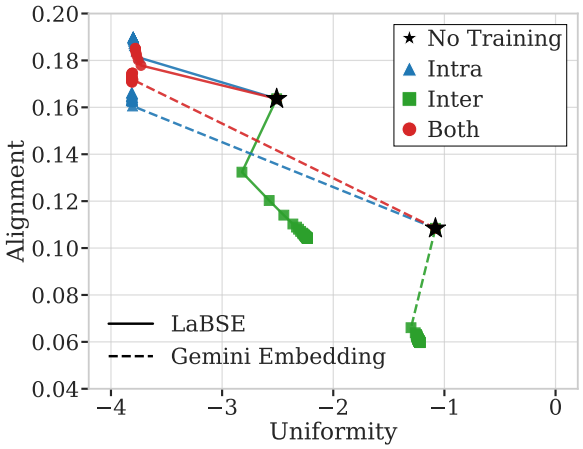

Figure 4: Alignment and Uniformity values at each training epoch on the SemEval-2017 task. "No Training" refers to the original sentence embedding. "Intra" and "Inter" denote intra-component and inter-component constraints, respectively, and "Both" indicates their combination.

5.2 Experimental Setup

To focus on similarity in meaning, we conduct the analysis using the SemEval-2017 STS task. For each language pair, 1,000 sentence pairs are randomly sampled from the validation data. Alignment and Uniformity are measured at each training epoch. As representative encoder-based and LLM-based models, we use LaBSE and Gemini Embedding, respectively. Embedding extraction methods, model configurations, hyperparameters, and training data are the same as described in Section 4.1.

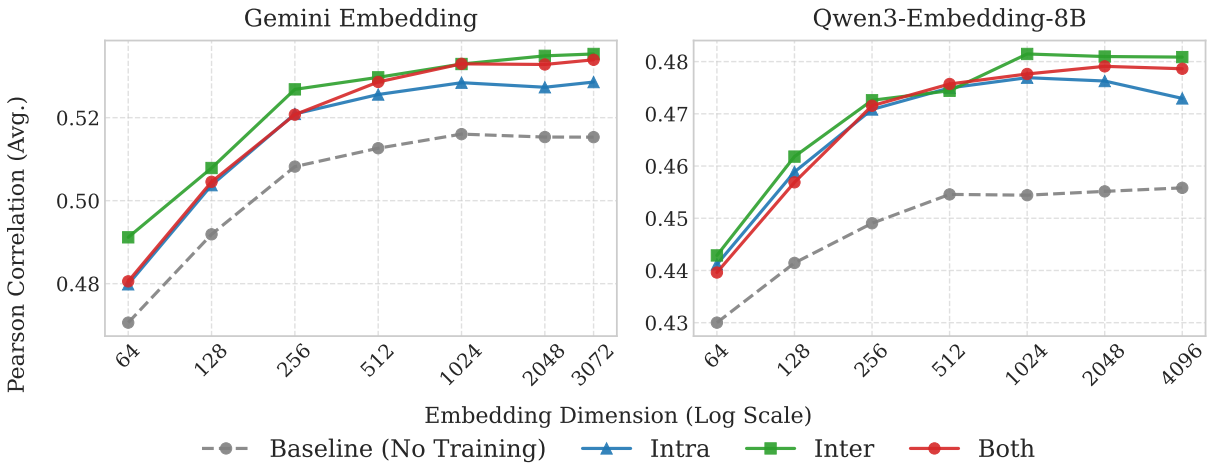

Figure 5: Changes in WMT20 performance (Avg.) as the embedding dimensionality is reduced. "Baseline (No Training)" refers to the original sentence embedding. "Intra" and "Inter" denote intra-component and inter-component constraints, respectively.

5.3 Results

The results are shown in Figure 4. For both models, training with the Intra-component constraint or with both constraints significantly decreases the Uniformity value, indicating that the embedding space is utilized more broadly. However, this is accompanied by a deterioration in Alignment. This behavior can be attributed to the second and third terms in the meaning embedding loss (Equation 3), which minimize the similarity between sentences with different meanings. These terms act as repulsive forces that push sentence embeddings apart, thereby improving Uniformity. In contrast, when using only the Inter-component constraint, Uniformity remains relatively stable, while Alignment improves substantially. This suggests that the cross-reconstruction framework enhances embedding separation and cross-lingual consistency without disturbing the global distribution of embeddings. Overall, these findings indicate that performance gains in encoder-based models are associated with improvements in Uniformity, whereas performance gains in LLM-based models are primarily associated with improvements in Alignment.

6 Analysis of Embedding Dimensionality

The previous analyses revealed architectural differences in the effectiveness of constraints. In this section, we focus on embedding dimensionality as one possible factor underlying these differences. As shown in Table 2, encoder-based models use lower-dimensional embeddings (768 and 1024 dimensions), whereas LLM-based models use much higher-dimensional embeddings (3072 and 4096 dimensions). This difference in dimensionality may affect the compatibility between model architecture and the proposed constraints. To investigate this possibility, we reduce the embedding dimensionality of LLM-based models and evaluate task performance under each constraint.

6.1 Experimental Setup

We use the WMT20 QE task, where embedding separation produces substantial performance differences. The LLM-based models evaluated are Gemini Embedding (3072 dimensions) and Qwen3-Embedding-8B (4096 dimensions). Both models adopt MRL (Kusupati et al., 2022), which enables dimensionality reduction by truncating leading dimensions while minimizing performance degradation. We progressively reduce the embedding dimensionality and train the MLP using the truncated embeddings. Other experimental settings are identical to those in Section 4.1.

6.2 Results

Figure 5 shows the average performance on the WMT20 QE task as embedding dimensionality is reduced. Across all evaluated dimensionalities, the Inter-component constraint consistently achieves the best performance for both models. In contrast, for the Intra-component constraint, performance gains tend to plateau or even diminish as dimensionality increases.

Combined with our geometric analysis, these results suggest that Inter-component constraints remain effective for LLM-based models regardless of

dimensionality, indicating that the observed differences stem from intrinsic architectural factors rather than embedding size. On the other hand, the Intra-component constraint, which directly manipulates pairwise similarities, appears to act as excessive regularization in high-dimensional LLM embedding spaces, potentially disrupting their inherent geometric structure.

7 Conclusion

In this study, we analyzed the relationship between loss functions and model architecture in methods that disentangle meaning and language representations from multilingual sentence embeddings. We categorized existing loss functions into Intra-component and Inter-component constraints, and examined their distinct effects on encoder-based and decoder-based models. Experimental results on the WMT20 QE and SemEval-2017 STS tasks showed that encoder-based models benefit most from the combination of both constraints, particularly the Intra-component constraint, whereas decoder-based models achieve the best performance using only the Inter-component constraint.

Acknowledgments

This work was supported by JST BOOST Program Japan Grant Number JPMJBY24036821.

References

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*, pages 1–14.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. 2022. DiffCSE: Difference-based Contrastive Learning for Sentence Embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4207–4218.

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*, pages 8440–8451.

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*, pages 4171–4186.

Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Rystrøm, Roman Solomatin, and 67 others. 2025. MMTEB: Massive Multilingual Text Embedding Benchmark. *arXiv:2502.13595*.

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*, pages 55–65.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT Sentence Embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 878–891.

Keita Fukushima, Tomoyuki Kajiwara, and Takashi Ninomiya. 2025. Reversible Disentanglement of Meaning and Language Representations from Multilingual Sentence Encoders. In *Proceedings of the 5th Workshop on Multilingual Representation Learning*, pages 265–270.

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.

Dayeon Ki, Cheonbok Park, and Hyunjoong Kim. 2024. Mitigating Semantic Leakage in Cross-lingual Embeddings via Orthogonality Constraint. In *Proceedings of the 9th Workshop on Representation Learning for NLP*, pages 256–273.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference for Learning Representations*.

Risa Kondo, Hiroki Yamauchi, Tomoyuki Kajiwara, Marie Katsurai, and Takashi Ninomiya. 2025. Domain Knowledge Distillation for Multilingual Sentence Encoders in Cross-lingual Sentence Similarity Estimation. In *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing*, pages 572–577.

Yuto Kuroda, Tomoyuki Kajiwara, Yuki Arase, and Takashi Ninomiya. 2022. Adversarial Training on

Disentangling Meaning and Language Representations for Unsupervised Quality Estimation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5240–5245.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. 2022. Matryoshka Representation Learning. In *Advances in Neural Information Processing Systems*, pages 30233–30249.

Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, Xiaoqi Ren, Shanfeng Zhang, Daniel Salz, Michael Boratko, Jay Han, Blair Chen, Shuo Huang, Vikram Rao, Paul Suganthan, and 28 others. 2025. Gemini Embedding: Generalizable Embeddings from Gemini. *arXiv:2503.07891*.

Chong Li, Shaonan Wang, Jiajun Zhang, and Chengqing Zong. 2024. Improving In-context Learning of Multilingual Generative Language Models with Cross-lingual Alignment. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 8058–8076.

Kanade Nonomura, Keita Fukushima, Risa Kondo, and Tomoyuki Kajiwara. 2026. Disentangling Meaning and Language Components in Diverse Multilingual Sentence Embeddings. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 Shared Task on Quality Estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764.

Nattapong Tiyajamorn, Tomoyuki Kajiwara, Yuki Arase, and Makoto Onizuka. 2021. Language-agnostic Representation from Multilingual Sentence Encoders for Cross-lingual Similarity Estimation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7764–7774.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual E5 Text Embeddings: A Technical Report. *arXiv:2402.05672*.

Tongzhou Wang and Phillip Isola. 2020. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *Proceedings of the 37th International Conference on Machine Learning*, pages 9929–9939.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. *arXiv:2506.05176*.