Dimensionality in Contrastive Sentence Embeddings

Alena Chan Matteo Bordoni Maria Garmonina Aksel Kretsinger-Walters

The goal is to represent each sentence by a vector in some m-dimensional space so that cosine similarity reflects semantic relevance. Our focus is to understand how the output dimension m affects encoder properties such as expected contrastive loss and alignment/uniformity on the unit sphere S^{m-1}. We aim to derive this relationship analytically and confirm it empirically.

Popular sentence-embedding models like Sentence-BERT can be viewed as two parts (Reimers & Gurevych, 2019):

1. **Bi-encoder architecture** The encoder maps a sentence with T tokens to a vector in $\mathbb{R}^m$ in three stages:

   Transformer encoding. A pre-trained encoder (e.g., BERT) maps each of the T tokens to a contextual vector in $\mathbb{R}^d$ (with $d = 768$ for BERT-base).

   Pooling. Aggregate the T token vectors into a single sentence vector $v \in \mathbb{R}^d$ (e.g., by taking the mean).

   Projection & normalization. Apply a learned linear map $W : \mathbb{R}^d \to \mathbb{R}^m$ to get $z = Wv$, then ℓ_2-normalize $\tilde{z} = z/\|z\|_2$ so dot product equals cosine.

2. **Contrastive learning objective.** We train with an InfoNCE loss over a minibatch. For anchor i and its positive i^+:

$$\mathcal{L}_i \;=\; -\log \frac{\exp\big(\mathrm{sim}(\tilde{z}_i, \tilde{z}_i^+)/\tau\big)}{\sum_j \exp\big(\mathrm{sim}(\tilde{z}_i, \tilde{z}_j)/\tau\big)}\,, \qquad \mathrm{sim} = \text{cosine}, \quad \tau > 0.$$

   Here $\tilde{z}_i$ is the normalized embedding of one view of a sentence, $\tilde{z}_i^+$ is its positive (e.g., an independently dropped-out view as in SimCSE; Gao et al., 2021), and $\tilde{z}_j$ ranges over in-batch negatives. The encoder parameters and the projection W are optimized by gradient descent to pull positives together and push negatives apart.

Our question is: How small can m be while preserving retrieval metrics, and what principled rule can guide choosing m? In practice, $d = 768$ is the BERT-base hidden size; many systems project to $m \in \{256, 384\}$ (and sometimes 128) for a good quality/size trade-off. Wang et al. (2023) report that m can be reduced to 128 using a two-step procedure with minimal loss on several STS/classification tasks, but they do not provide a general theory relating m to encoding performance.

On the theory side, we would like to consider the above question using the framework proposed by Wang and Isola 2020. They argue that contrastive learning on unit-normalized embeddings implicitly balances *alignment*, which brings positive pairs together, and *uniformity*, which spreads embeddings roughly uniformly on S^{m-1}. There is no explicit discussion of dimensionality in Wang and Isola's paper, but it seems possible to consider the question of dimensionality in the context of their results. Another potential theoretical angle is the Johnson-Lindenstrauss lemma.

On the experimental side, one thing we could do is to validate any predicted relationship between m and retrieval metrics/alignment/uniformity. Another thing we could try is to compare two ways of reducing m. The first way, which is similar to what Wang el al. 2023 does, is to set the projection to target $m \in \{128, 200, 256, 384\}$ and fine-tune encoder with the contrastive loss function. The second way is to take a trained encoder and do standard dimensionality reduction techniques on the output vector; this is not tried by Wang el al. 2023. An empirical question is how far m can go before the retrieval quality seriously suffers.

To score retrieval quality, we can use standard semantic textual similarity (STS) tasks (e.g., STS-B, SICK-R) and report Spearman correlation between cosine similarities of sentence pairs and human annotations. Some competing methods to compare against and benchmark include:

- N-Gram BOW (TD-IDF)
- Latent Semantic Analysis (LSA)
- GloVe /Word2Vec embeddings
- FastText embeddings
- Doc-2Vec
- Universal Sentence Encoder (Cer et al., 2018)
- SBERT and SimCSE (as described above)

To produce data for the training and analysis, we can enrich the standard STS datasets with synthetic data. For example, we can take a sentence and produce positive pairs by independently applying dropout, synonym replacement, and back-translation. Negatives can be sampled from the rest of the batch.

References

[1] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP) and the 9th International Joint Conference on Natural Language Processing (IJCNLP)*, pp. 3982–3992, 2019. `https://aclanthology.org/D19-1410`.

[2] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6894–6910, 2021. `https://aclanthology.org/2021.emnlp-main.552`.

[3] Tongzhou Wang and Phillip Isola. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. *Proceedings of the 37th International Conference on Machine Learning (ICML)*, PMLR 119:9929–9939, 2020. `https://proceedings.mlr.press/v119/wang20k.html`.

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