Evaluating Unsupervised Dimensionality Reduction Methods for Pretrained Sentence Embeddings

Gaifan Zhang[1], Yi Zhou[2], Danushka Bollegala[3]

Columbia University[1], Cardiff University[2], University of Liverpool[3]

LREC-COLING 2024

Problems with sentence embeddings

1. storing pre-computed sentence embeddings requires larger memory/disk space

2. the computation time of the inner-products between two sentence embeddings increases linearly with the dimensionality of the embedding

➢ trade-off between the dimensionality and the accuracy of sentence embeddings

Motivation

Can we reduce the dimensionality of pre-computed sentence embeddings without significantly sacrificing the performance in downstream tasks that use those *dimensionality-reduced* sentence embeddings?

Post-Processing Dimensionality Reduction

- Given $\mathcal{D}_{\text{train}} = \{s_1, s_2, \ldots, s_n\}$, $M(s) = \vec{s} \in \mathbb{R}^d$
- Learn $\boldsymbol{f}: \mathbb{R}^{d} \rightarrow \mathbb{R}^{d'}$ where $d' < d$

where M is the pretrained sentence encoder, $\mathcal{D}_{\text{train}}$ is a set of train sentences, d is the original dimensionality, and d' is the reduced dimensionality

Evaluation Framework

- **5** unsupervised DR methods
  - Principal Component Analysis (**PCA**)
  - Kernel PCA (**KPCA**)
  - Gaussian Random Projection (**GRP**)
  - Autoencoder
  - Truncated Singular Value Decomposition (**SVD**)

Evaluation Framework

- **6** sentence encoders
  - all-mpnet-base-v2 (**mpnet**)
  - stsb-bert-base (**sbert-b**)
  - msmarco-roberta-base-v2 (**roberta**)
  - paraphrase-xlm-r-multilingual-v1 (**xml-r**)
  - stsb-bert-large (**sbert-l**)
  - sup-simcse-roberta-large (**simcse**)

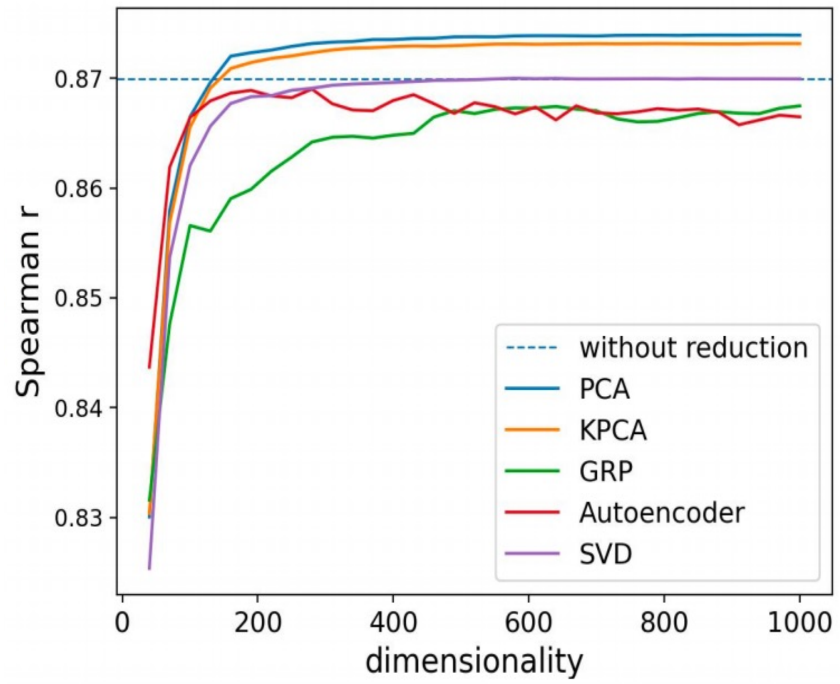
Evaluation Framework

- **3** downstream tasks
  - Semantic Textual Similarity Prediction (STS-B)
  - Question Classification (TREC)
  - Textual Entailment (SICK-E)

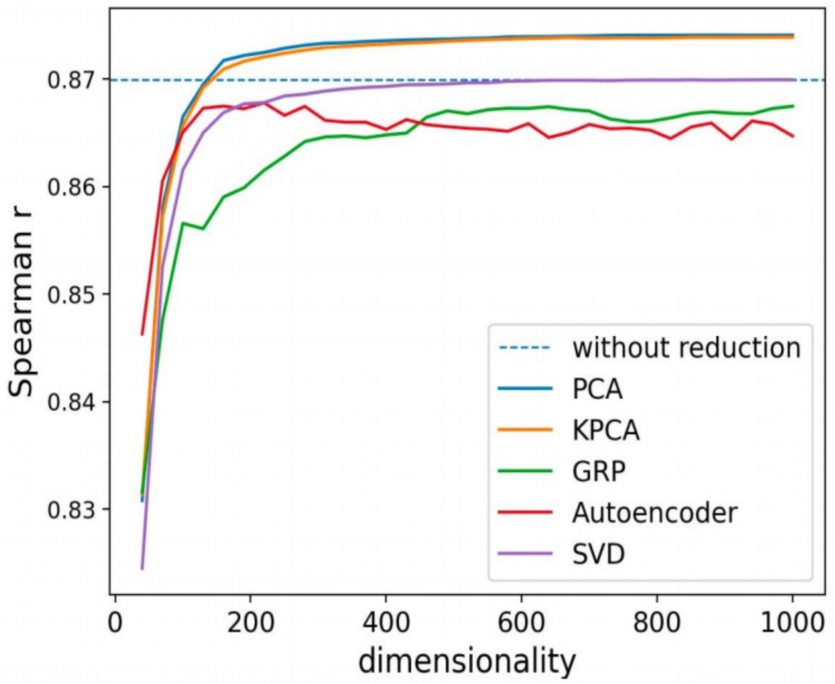
Evaluation Framework

- **2** settings
  - transduction (specificity)
  - induction (generalization)
- **2** evaluation indicators
  - accuracy
  - time cost

Results – **STS-B** Task Performance



transductive

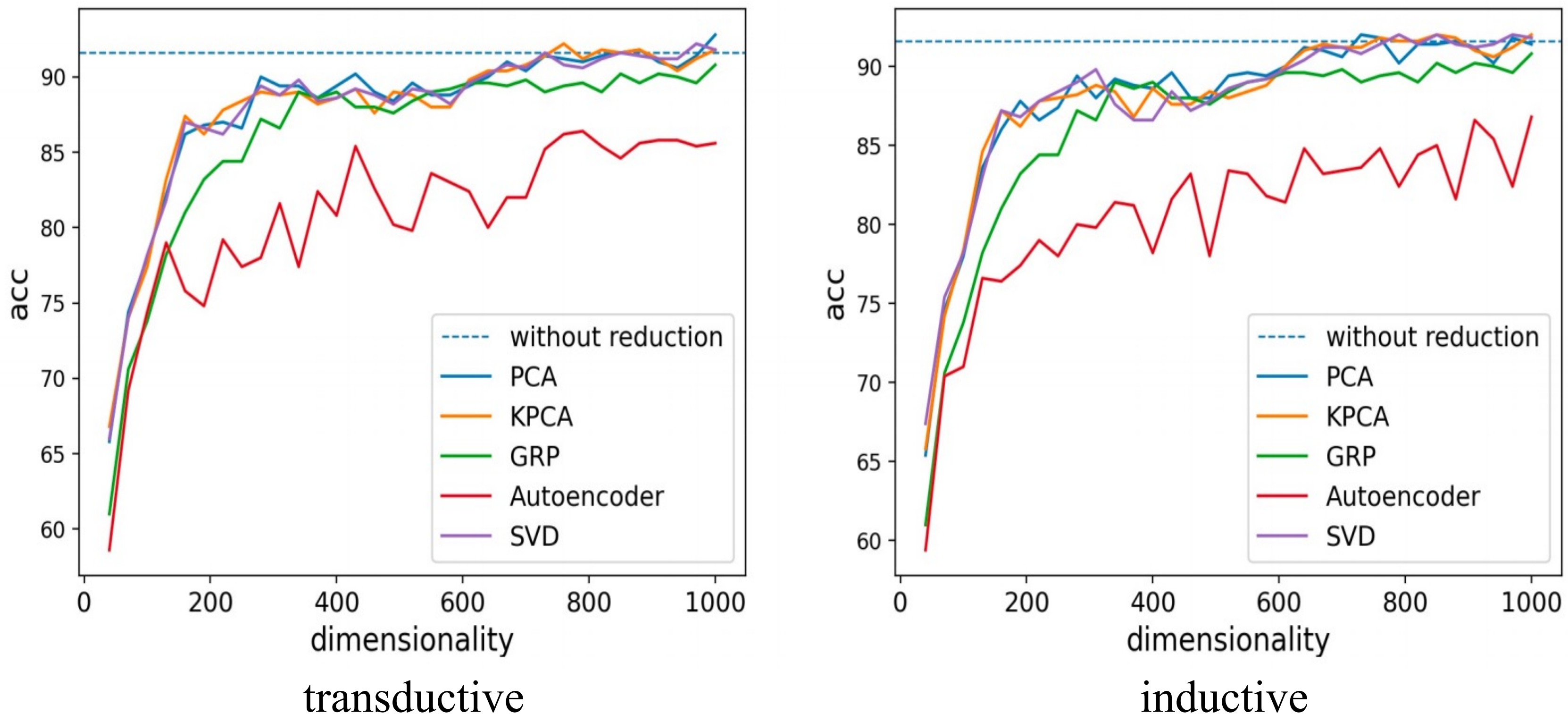


inductive

Performance of the original **simcse** sentence embeddings and its reduced versions

- high performance of PCA and KPCA
- same performance of GRP in both transductive and inductive settings
- reduced embeddings improves performance in some cases

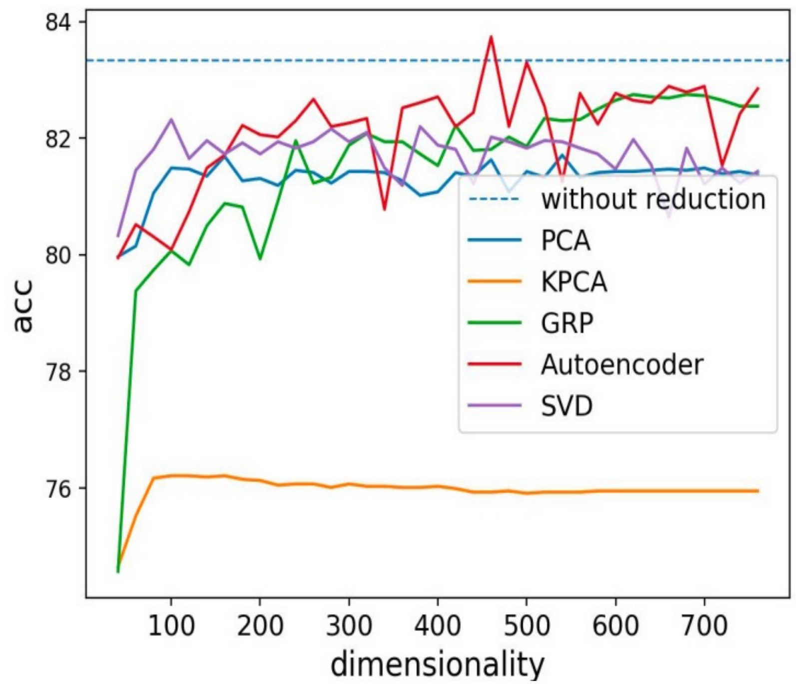
Results – **TREC** Task Performance



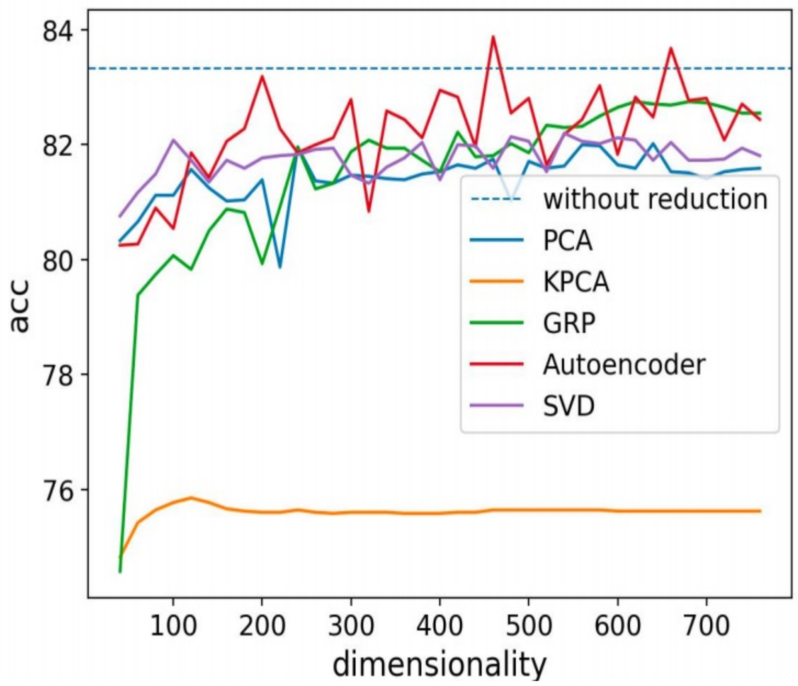
Performance of the original **simcse** sentence embeddings and its reduced versions

- similar performance of PCA, KPCA and SVD
- unstable and poor performance of autoencoder

Results – **SICK-E** Task Performance



transductive

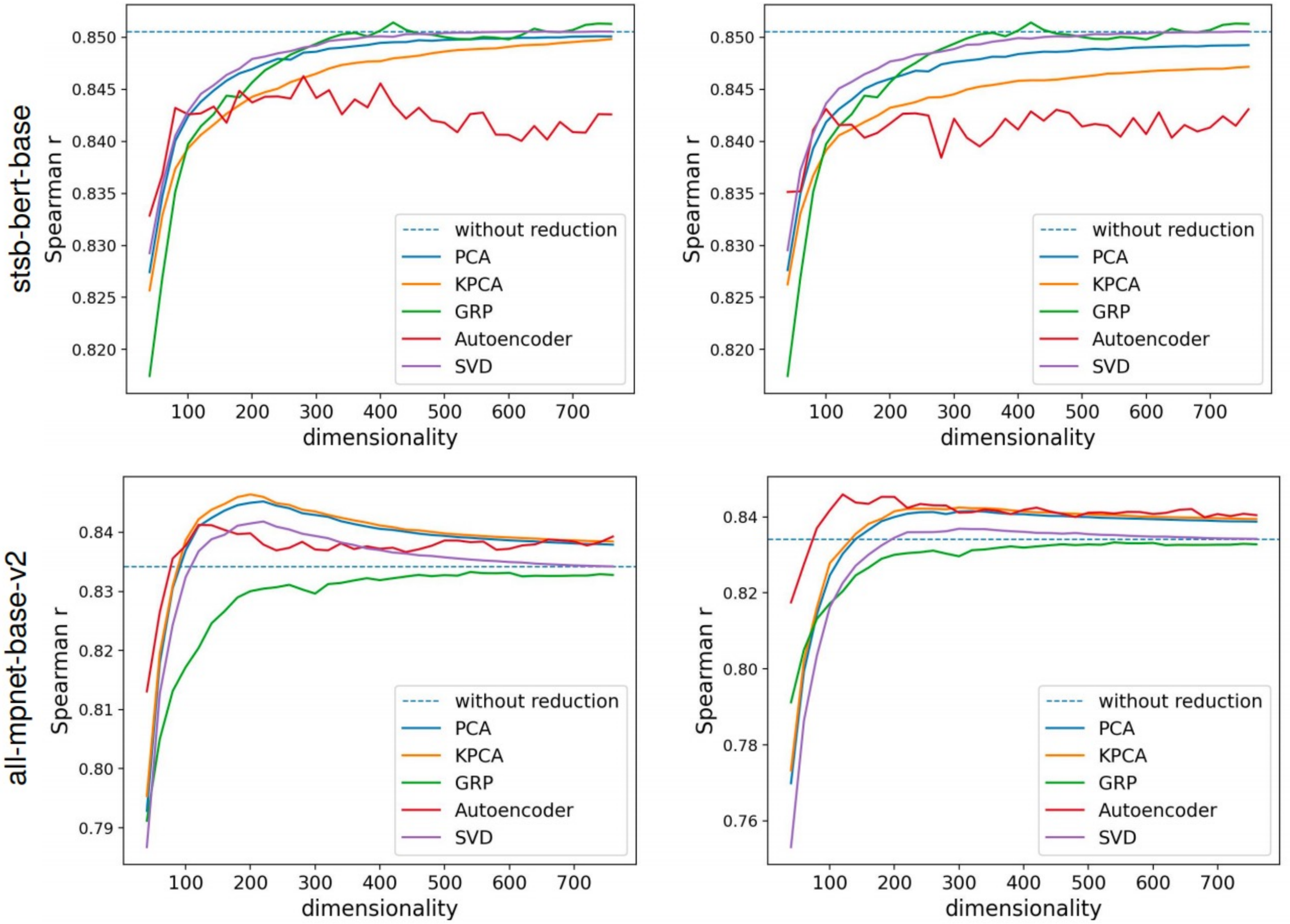


inductive

Performance of the original **mpnet** sentence embeddings and its reduced versions

- poor performance of KPCA
- relatively good performance of autoencoder in this case

Results – Performance on different encoders



- diverse performance of same method on different sentence encoders for the same task

Performance of sentence embeddings on STS-B

Results – Training and inference times

Method	Training Time (s)	Inference time (s)
PCA	2.08	**0.0049**
KPCA	37.98	0.7883
SVD	2.57	0.0089
Autoencoder	101.16	0.1479
GRP	**0.03**	0.0080

Training and inference times measured on the test set of STS-B under the inductive setting, with **mpnet** reduced to 300 dimensions.

- fast training and inference time of GRP, PCA and SVD (matrix projection)
- autoencoders and KPCA are both slow to train and infer with
  - backpropagation and iterations of autoencoders
  - kernel matrix of KPCA

Conclusion

- We evaluated unsupervised dimensionality reduction methods for pre-trained sentence embeddings using multiple NLP tasks and benchmarks under transductive and inductive settings.
- PCA performs consistently well across encoders and tasks. PCA can reduce the dimensionality by almost 50%, without incurring a significant loss in performance.
- Reducing the dimensionality improves performance over the original high-dimensional sentence embeddings produced by some PLMs in some tasks.