

[![Logo](../../_static/logo.png)](../../index.html)

Getting Started

- * [Installation](../../docs/installation.html)

- * [Install with pip](../../docs/installation.html#install-with-pip)

- * [Install with Conda](../../docs/installation.html#install-with-conda)

- * [Install from Source](../../docs/installation.html#install-from-source)

- * [Editable Install](../../docs/installation.html#editable-install)

- * [Install PyTorch with CUDA support](../../docs/installation.html#install-pytorch-with-cuda-support)

- * [Quickstart](../../docs/quickstart.html)

- * [Sentence Transformer](../../docs/quickstart.html#sentence-transformer)

- * [Cross Encoder](../../docs/quickstart.html#cross-encoder)

- * [Next Steps](../../docs/quickstart.html#next-steps)

Sentence Transformer

- * [Usage](../../docs/sentence_transformer/usage/usage.html)

- * [Computing Embeddings](../../applications/computing-embeddings/README.html)

- * [Initializing a Sentence Transformer Model](../../applications/computing-embeddings/README.html#initializing-a-sentence-transformer-model)

- * [Calculating Embeddings](../../applications/computing-embeddings/README.html#calculating-embeddings)

- * [Prompt Templates](../../applications/computing-embeddings/README.html#prompt-templates)

- * [Input Sequence Length](../../applications/computing-embeddings/README.html#id1)

* [Multi-Process / Multi-GPU

Encoding](../../applications/computing-embeddings/README.html#multi-process-multi-gpu-encoding)

* [Semantic Textual

Similarity](../../docs/sentence_transformer/usage/semantic_textual_similarity.html)

* [Similarity

Calculation](../../docs/sentence_transformer/usage/semantic_textual_similarity.html#similarity-calculation)

* [Semantic Search](../../applications/semantic-search/README.html)

* [Background](../../applications/semantic-search/README.html#background)

* [Symmetric vs. Asymmetric Semantic

Search](../../applications/semantic-search/README.html#symmetric-vs-asymmetric-semantic-search)

* [Manual

Implementation](../../applications/semantic-search/README.html#manual-implementation)

* [Optimized

Implementation](../../applications/semantic-search/README.html#optimized-implementation)

* [Speed Optimization](../../applications/semantic-search/README.html#speed-optimization)

* [Elasticsearch](../../applications/semantic-search/README.html#elasticsearch)

* [Approximate Nearest

Neighbor](../../applications/semantic-search/README.html#approximate-nearest-neighbor)

* [Retrieve & Re-Rank](../../applications/semantic-search/README.html#retrieve-re-rank)

* [Examples](../../applications/semantic-search/README.html#examples)

* [Retrieve & Re-Rank](../../applications/retrieve_rerank/README.html)

* [Retrieve & Re-Rank

Pipeline](../../applications/retrieve_rerank/README.html#retrieve-re-rank-pipeline)

* [Retrieval: Bi-Encoder](../../applications/retrieve_rerank/README.html#retrieval-bi-encoder)

[Cross-Encoder\]\(../../applications/retrieve_rerank/README.html#re-ranker-cross-encoder\)](#)
 * [\[Example Scripts\]\(../../applications/retrieve_rerank/README.html#example-scripts\)](#)
 * [\[Pre-trained Bi-Encoders \(Retrieval\)\]\(../../applications/retrieve_rerank/README.html#pre-trained-bi-encoders-retrieval\)](#)
 * [\[Pre-trained Cross-Encoders \(Re-Ranker\)\]\(../../applications/retrieve_rerank/README.html#pre-trained-cross-encoders-re-ranker\)](#)
 * [\[Clustering\]\(../../applications/clustering/README.html\)](#)
 * [\[k-Means\]\(../../applications/clustering/README.html#k-means\)](#)
 * [\[Agglomerative Clustering\]\(../../applications/clustering/README.html#agglomerative-clustering\)](#)
 * [\[Fast Clustering\]\(../../applications/clustering/README.html#fast-clustering\)](#)
 * [\[Topic Modeling\]\(../../applications/clustering/README.html#topic-modeling\)](#)
 * [\[Paraphrase Mining\]\(../../applications/paraphrase-mining/README.html\)](#)
 * [\[paraphrase_minning\(\)\]\(../../applications/paraphrase-mining/README.html#sentence_transformers.util.paraphrase_minning\)](#)
 * [\[Translated Sentence Mining\]\(../../applications/parallel-sentence-mining/README.html\)](#)
 * [\[Margin Based Mining\]\(../../applications/parallel-sentence-mining/README.html#margin-based-mining\)](#)
 * [\[Examples\]\(../../applications/parallel-sentence-mining/README.html#examples\)](#)
 * [\[Image Search\]\(../../applications/image-search/README.html\)](#)
 * [\[Installation\]\(../../applications/image-search/README.html#installation\)](#)
 * [\[Usage\]\(../../applications/image-search/README.html#usage\)](#)
 * [\[Examples\]\(../../applications/image-search/README.html#examples\)](#)
 * [\[Embedding Quantization\]\(../../applications/embedding-quantization/README.html\)](#)
 * [\[Binary Quantization\]\(../../applications/embedding-quantization/README.html#binary-quantization\)](#)

[Quantization\]\(../../applications/embedding-quantization/README.html#scalar-int8-quantization\)](#)

[\[Additional extensions\]\(../../applications/embedding-quantization/README.html#additional-extensions\)](#)

- * [\[Demo\]\(../../applications/embedding-quantization/README.html#demo\)](#)
- * [\[Try it yourself\]\(../../applications/embedding-quantization/README.html#try-it-yourself\)](#)
- * [\[Speeding up Inference\]\(../../docs/sentence_transformer/usage/efficiency.html\)](#)
- * [\[PyTorch\]\(../../docs/sentence_transformer/usage/efficiency.html#pytorch\)](#)
- * [\[ONNX\]\(../../docs/sentence_transformer/usage/efficiency.html#onnx\)](#)
- * [\[OpenVINO\]\(../../docs/sentence_transformer/usage/efficiency.html#openvino\)](#)
- * [\[Benchmarks\]\(../../docs/sentence_transformer/usage/efficiency.html#benchmarks\)](#)
- * [\[Creating Custom Models\]\(../../docs/sentence_transformer/usage/custom_models.html\)](#)

[* \[Structure of Sentence Transformer Models\]\(../../docs/sentence_transformer/usage/custom_models.html#structure-of-sentence-transformer-models\)](#)

[* \[Sentence Transformer Model from a Transformers Model\]\(../../docs/sentence_transformer/usage/custom_models.html#sentence-transformer-model-from-a-transformers-model\)](#)

- * [\[Pretrained Models\]\(../../docs/sentence_transformer/pretrained_models.html\)](#)
- * [\[Original Models\]\(../../docs/sentence_transformer/pretrained_models.html#original-models\)](#)

[* \[Semantic Search Models\]\(../../docs/sentence_transformer/pretrained_models.html#semantic-search-models\)](#)

- * [\[Multi-QA Models\]\(../../docs/sentence_transformer/pretrained_models.html#multi-qa-models\)](#)

[* \[MSMARCO Passage Models\]\(../../docs/sentence_transformer/pretrained_models.html#msmarco-passage-models\)](#)

[* \[Multilingual Models\]\(../../docs/sentence_transformer/pretrained_models.html#multilingual-models\)](#)

[* \[Semantic Similarity Models\]\(../../docs/sentence_transformer/pretrained_models.html#semantic-similarity-models\)](#)
[* \[Bitext Mining\]\(../../docs/sentence_transformer/pretrained_models.html#bitext-mining\)](#)
[* \[Image & Text-Models\]\(../../docs/sentence_transformer/pretrained_models.html#image-text-models\)](#)
[* \[INSTRUCTOR models\]\(../../docs/sentence_transformer/pretrained_models.html#instructor-models\)](#)
[* \[Scientific Similarity Models\]\(../../docs/sentence_transformer/pretrained_models.html#scientific-similarity-models\)](#)
[* \[Training Overview\]\(../../docs/sentence_transformer/training_overview.html\)](#)
[* \[Why Finetune?\]\(../../docs/sentence_transformer/training_overview.html#why-finetune\)](#)
[* \[Training Components\]\(../../docs/sentence_transformer/training_overview.html#training-components\)](#)
[* \[Dataset\]\(../../docs/sentence_transformer/training_overview.html#dataset\)](#)
[* \[Dataset Format\]\(../../docs/sentence_transformer/training_overview.html#dataset-format\)](#)
[* \[Loss Function\]\(../../docs/sentence_transformer/training_overview.html#loss-function\)](#)
[* \[Training Arguments\]\(../../docs/sentence_transformer/training_overview.html#training-arguments\)](#)
[* \[Evaluator\]\(../../docs/sentence_transformer/training_overview.html#evaluator\)](#)
[* \[Trainer\]\(../../docs/sentence_transformer/training_overview.html#trainer\)](#)
[* \[Callbacks\]\(../../docs/sentence_transformer/training_overview.html#callbacks\)](#)
[* \[Multi-Dataset Training\]\(../../docs/sentence_transformer/training_overview.html#multi-dataset-training\)](#)
[* \[Deprecated Training\]\(../../docs/sentence_transformer/training_overview.html#deprecated-training\)](#)
[* \[Best Base Embedding Models\]\(../../docs/sentence_transformer/training_overview.html#best-base-embedding-models\)](#)

- * [Dataset Overview](../../docs/sentence_transformer/dataset_overview.html)
- * [Datasets on the Hugging Face Hub](../../docs/sentence_transformer/dataset_overview.html#datasets-on-the-hugging-face-hub)
- * [Pre-existing Datasets](../../docs/sentence_transformer/dataset_overview.html#pre-existing-datasets)
- * [Loss Overview](../../docs/sentence_transformer/loss_overview.html)
- * [Loss modifiers](../../docs/sentence_transformer/loss_overview.html#loss-modifiers)
- * [Distillation](../../docs/sentence_transformer/loss_overview.html#distillation)
- * [Commonly used Loss Functions](../../docs/sentence_transformer/loss_overview.html#commonly-used-loss-functions)
- * [Custom Loss Functions](../../docs/sentence_transformer/loss_overview.html#custom-loss-functions)
- * [Training Examples](../../docs/sentence_transformer/training/examples.html)
- * [Semantic Textual Similarity](../sts/README.html)
- * [Training data](../sts/README.html#training-data)
- * [Loss Function](../sts/README.html#loss-function)
- * [Natural Language Inference](../nli/README.html)
- * [Data](../nli/README.html#data)
- * [SoftmaxLoss](../nli/README.html#softmaxloss)
- * [MultipleNegativesRankingLoss](../nli/README.html#multiplenegativesrankingloss)
- * [Paraphrase Data](../paraphrases/README.html)
- * [Pre-Trained Models](../paraphrases/README.html#pre-trained-models)
- * [Quora Duplicate Questions](../quora_duplicate_questions/README.html)
- * [Training](../quora_duplicate_questions/README.html#training)
- * [MultipleNegativesRankingLoss](../quora_duplicate_questions/README.html#multiplenegativesrankingloss)

- * [Pretrained Models](../quora_duplicate_questions/README.html#pretrained-models)
- * [MS MARCO](../ms_marco/README.html)
- * [Bi-Encoder](../ms_marco/README.html#bi-encoder)
- * [Matryoshka Embeddings](../matryoshka/README.html)
- * [Use Cases](../matryoshka/README.html#use-cases)
- * [Results](../matryoshka/README.html#results)
- * [Training](../matryoshka/README.html#training)
- * [Inference](../matryoshka/README.html#inference)
- * [Code Examples](../matryoshka/README.html#code-examples)
- * [Adaptive Layers](../adaptive_layer/README.html)
- * [Use Cases](../adaptive_layer/README.html#use-cases)
- * [Results](../adaptive_layer/README.html#results)
- * [Training](../adaptive_layer/README.html#training)
- * [Inference](../adaptive_layer/README.html#inference)
- * [Code Examples](../adaptive_layer/README.html#code-examples)
- * [Multilingual Models](../multilingual/README.html)
- * [Extend your own models](../multilingual/README.html#extend-your-own-models)
- * [Training](../multilingual/README.html#training)
- * [Datasets](../multilingual/README.html#datasets)
- * [Sources for Training Data](../multilingual/README.html#sources-for-training-data)
- * [Evaluation](../multilingual/README.html#evaluation)
- * [Available Pre-trained Models](../multilingual/README.html#available-pre-trained-models)
- * [Usage](../multilingual/README.html#usage)
- * [Performance](../multilingual/README.html#performance)
- * [Citation](../multilingual/README.html#citation)
- * Model Distillation
- * Knowledge Distillation

- * Speed - Performance Trade-Off
- * Dimensionality Reduction
- * Quantization
- * [Augmented SBERT](../data_augmentation/README.html)
- * [Motivation](../data_augmentation/README.html#motivation)
- * [Extend to your own datasets](../data_augmentation/README.html#extend-to-your-own-datasets)
- * [Methodology](../data_augmentation/README.html#methodology)
 - * [Scenario 1: Limited or small annotated datasets (few labeled sentence-pairs)](../data_augmentation/README.html#scenario-1-limited-or-small-annotated-dataset-s-few-labeled-sentence-pairs)
 - * [Scenario 2: No annotated datasets (Only unlabeled sentence-pairs)](../data_augmentation/README.html#scenario-2-no-annotated-datasets-only-unlabeled-sentence-pairs)
- * [Training](../data_augmentation/README.html#training)
- * [Citation](../data_augmentation/README.html#citation)
- * [Training with Prompts](../prompts/README.html)
- * [What are Prompts?](../prompts/README.html#what-are-prompts)
 - * [Why would we train with Prompts?](../prompts/README.html#why-would-we-train-with-prompts)
 - * [How do we train with Prompts?](../prompts/README.html#how-do-we-train-with-prompts)
- * [Training with PEFT Adapters](../peft/README.html)
 - * [Compatibility Methods](../peft/README.html#compatibility-methods)
 - * [Adding a New Adapter](../peft/README.html#adding-a-new-adapter)
 - * [Loading a Pretrained Adapter](../peft/README.html#loading-a-pretrained-adapter)
 - * [Training Script](../peft/README.html#training-script)
- * [Unsupervised Learning](../unsupervised_learning/README.html)

* [TSDAE](../../unsupervised_learning/README.html#tsdae)

* [SimCSE](../../unsupervised_learning/README.html#simcse)

* [CT](../../unsupervised_learning/README.html#ct)

* [CT (In-Batch Negative Sampling)](../../unsupervised_learning/README.html#ct-in-batch-negative-sampling)

* [Masked Language Model (MLM)](../../unsupervised_learning/README.html#masked-language-model-mlm)

* [GenQ](../../unsupervised_learning/README.html#genq)

* [GPL](../../unsupervised_learning/README.html#gpl)

* [Performance Comparison](../../unsupervised_learning/README.html#performance-comparison)

* [Domain Adaptation](../../domain_adaptation/README.html)

* [Domain Adaptation vs. Unsupervised Learning](../../domain_adaptation/README.html#domain-adaptation-vs-unsupervised-learning)

* [Adaptive Pre-Training](../../domain_adaptation/README.html#adaptive-pre-training)

* [GPL: Generative Pseudo-Labeling](../../domain_adaptation/README.html#gpl-generative-pseudo-labeling)

* [Hyperparameter Optimization](../hpo/README.html)

* [HPO Components](../hpo/README.html#hpo-components)

* [Putting It All Together](../hpo/README.html#putting-it-all-together)

* [Example Scripts](../hpo/README.html#example-scripts)

* [Distributed Training](../../docs/sentence_transformer/training/distributed.html)

* [Comparison](../../docs/sentence_transformer/training/distributed.html#comparison)

* [FSDP](../../docs/sentence_transformer/training/distributed.html#fsdp)

Cross Encoder

* [Usage](../../docs/cross_encoder/usage/usage.html)

* [Retrieve & Re-Rank](../../applications/retrieve_rerank/README.html)

* [Retrieve & Re-Rank Pipeline](../../applications/retrieve_rerank/README.html#retrieve-re-rank-pipeline)

* [Retrieval: Bi-Encoder](../../applications/retrieve_rerank/README.html#retrieval-bi-encoder)

* [Re-Ranker: Cross-Encoder](../../applications/retrieve_rerank/README.html#re-ranker-cross-encoder)

* [Example Scripts](../../applications/retrieve_rerank/README.html#example-scripts)

* [Pre-trained Bi-Encoders (Retrieval)](../../applications/retrieve_rerank/README.html#pre-trained-bi-encoders-retrieval)

* [Pre-trained Cross-Encoders (Re-Ranker)](../../applications/retrieve_rerank/README.html#pre-trained-cross-encoders-re-ranker)

* [Pretrained Models](../../docs/cross_encoder/pretrained_models.html)

* [MS MARCO](../../docs/cross_encoder/pretrained_models.html#ms-marco)

* [SQuAD (QNLI)](../../docs/cross_encoder/pretrained_models.html#squad-qnli)

* [STSbenchmark](../../docs/cross_encoder/pretrained_models.html#stsbenchmark)

* [Quora Duplicate Questions](../../docs/cross_encoder/pretrained_models.html#quora-duplicate-questions)

* [NLI](../../docs/cross_encoder/pretrained_models.html#nli)

* [Community Models](../../docs/cross_encoder/pretrained_models.html#community-models)

* [Training Overview](../../docs/cross_encoder/training_overview.html)

* [Training Examples](../../docs/cross_encoder/training/examples.html)

* [MS MARCO](../ms_marco/cross_encoder_README.html)

* [Cross-Encoder](../ms_marco/cross_encoder_README.html#cross-encoder)

* [Cross-Encoder Knowledge Distillation](../ms_marco/cross_encoder_README.html#cross-encoder-knowledge-distillation)

Package Reference

* [Sentence Transformer](../../docs/package_reference/sentence_transformer/index.html)

*

[SentenceTransformer](../../docs/package_reference/sentence_transformer/SentenceTransformer.html)

*

[SentenceTransformer](../../docs/package_reference/sentence_transformer/SentenceTransformer.html#id1)

*

[SentenceTransformerModelCardData](../../docs/package_reference/sentence_transformer/SentenceTransformer.html#sentencetransformermodelcarddata)

*

[SimilarityFunction](../../docs/package_reference/sentence_transformer/SentenceTransformer.html#similarityfunction)

* [Trainer](../../docs/package_reference/sentence_transformer/trainer.html)

*

[SentenceTransformerTrainer](../../docs/package_reference/sentence_transformer/trainer.html#sentencetransformertrainer)

* [Training Arguments](../../docs/package_reference/sentence_transformer/training_args.html)

*

[SentenceTransformerTrainingArguments](../../docs/package_reference/sentence_transformer/training_args.html#sentencetransformertrainingarguments)

* [Losses](../../docs/package_reference/sentence_transformer/losses.html)

*

[BatchAllTripletLoss](../../docs/package_reference/sentence_transformer/losses.html#batchalltripletloss)

*

[BatchHardSoftMarginTripletLoss](../../docs/package_reference/sentence_transformer/losses.html#batchhardsoftmargintripletloss)

*

[BatchHardTripletLoss](../../docs/package_reference/sentence_transformer/losses.html#batchhardtripletloss)

*

[BatchSemiHardTripletLoss](../../docs/package_reference/sentence_transformer/losses.html#batchsemi-hardtripletloss)

*

[ContrastiveLoss](../../docs/package_reference/sentence_transformer/losses.html#contrastiveloss)

*

[OnlineContrastiveLoss](../../docs/package_reference/sentence_transformer/losses.html#onlinecontrastiveloss)

*

[ContrastiveTensionLoss](../../docs/package_reference/sentence_transformer/losses.html#contrastivetensionloss)

*

[ContrastiveTensionLossInBatchNegatives](../../docs/package_reference/sentence_transformer/losses.html#contrastivetensionlossinbatchnegatives)

* [CoSENTLoss](../../docs/package_reference/sentence_transformer/losses.html#cosentloss)

* [AngleLoss](../../docs/package_reference/sentence_transformer/losses.html#angleloss)

*

[CosineSimilarityLoss](../../docs/package_reference/sentence_transformer/losses.html#cosinesimilarityloss)

*

[DenoisingAutoEncoderLoss](../../docs/package_reference/sentence_transformer/losses.html#denoisingautoencoderloss)

osingautoencoderloss)

*

[GISTEmbedLoss](../../docs/package_reference/sentence_transformer/losses.html#gistembedloss
)

*

[CachedGISTEmbedLoss](../../docs/package_reference/sentence_transformer/losses.html#cachedgistembedloss)

* [MSELoss](../../docs/package_reference/sentence_transformer/losses.html#mseloss)

*

[MarginMSELoss](../../docs/package_reference/sentence_transformer/losses.html#marginmseloss
)

*

[MatryoshkaLoss](../../docs/package_reference/sentence_transformer/losses.html#matryoshkaloss
)

*

[Matryoshka2dLoss](../../docs/package_reference/sentence_transformer/losses.html#matryoshka2dloss)

*

[AdaptiveLayerLoss](../../docs/package_reference/sentence_transformer/losses.html#adaptivelayerloss)

*

[MegaBatchMarginLoss](../../docs/package_reference/sentence_transformer/losses.html#megabatchmarginloss)

*

[MultipleNegativesRankingLoss](../../docs/package_reference/sentence_transformer/losses.html#multiplenegativesrankingloss)

*

[CachedMultipleNegativesRankingLoss](../../docs/package_reference/sentence_transformer/losses.html#cachedmultiplenegativesrankingloss)

*

[MultipleNegativesSymmetricRankingLoss](../../docs/package_reference/sentence_transformer/losses.html#multiplenegativessymmetricrankingloss)

*

[CachedMultipleNegativesSymmetricRankingLoss](../../docs/package_reference/sentence_transformer/losses.html#cachedmultiplenegativessymmetricrankingloss)

* [SoftmaxLoss](../../docs/package_reference/sentence_transformer/losses.html#softmaxloss)

* [TripletLoss](../../docs/package_reference/sentence_transformer/losses.html#tripletloss)

* [Samplers](../../docs/package_reference/sentence_transformer/sampler.html)

*

[BatchSamplers](../../docs/package_reference/sentence_transformer/sampler.html#batchsamplers)

*

[MultiDatasetBatchSamplers](../../docs/package_reference/sentence_transformer/sampler.html#multidatasetbatchsamplers)

* [Evaluation](../../docs/package_reference/sentence_transformer/evaluation.html)

*

[BinaryClassificationEvaluator](../../docs/package_reference/sentence_transformer/evaluation.html#binaryclassificationevaluator)

*

[EmbeddingSimilarityEvaluator](../../docs/package_reference/sentence_transformer/evaluation.html#embeddingsimilarityevaluator)

*

[InformationRetrievalEvaluator](../../docs/package_reference/sentence_transformer/evaluation.html#informationretrievalevaluator)

*

[NanoBEIREvaluator](../../docs/package_reference/sentence_transformer/evaluation.html#nanobe
irevaluator)

*

[MSEEvaluator](../../docs/package_reference/sentence_transformer/evaluation.html#mseevaluator
)

*

[ParaphraseMiningEvaluator](../../docs/package_reference/sentence_transformer/evaluation.html#
paraphraseminingevaluator)

*

[RerankingEvaluator](../../docs/package_reference/sentence_transformer/evaluation.html#rerankin
gevaluator)

*

[SentenceEvaluator](../../docs/package_reference/sentence_transformer/evaluation.html#sentenc
eevaluator)

*

[SequentialEvaluator](../../docs/package_reference/sentence_transformer/evaluation.html#sequen
tiaevaluator)

*

[TranslationEvaluator](../../docs/package_reference/sentence_transformer/evaluation.html#translat
ionevaluator)

*

[TripletEvaluator](../../docs/package_reference/sentence_transformer/evaluation.html#tripletevalua
tor)

* [Datasets](../../docs/package_reference/sentence_transformer/datasets.html)

*

[ParallelSentencesDataset](../../docs/package_reference/sentence_transformer/datasets.html#par

allelsentencesdataset)

*

[SentenceLabelDataset](../../docs/package_reference/sentence_transformer/datasets.html#sentence-label-dataset)

*

[DenoisingAutoEncoderDataset](../../docs/package_reference/sentence_transformer/datasets.html#denoising-auto-encoder-dataset)

*

[NoDuplicatesDataLoader](../../docs/package_reference/sentence_transformer/datasets.html#no-duplicates-data-loader)

* [Models](../../docs/package_reference/sentence_transformer/models.html)

*

[Main

Classes](../../docs/package_reference/sentence_transformer/models.html#main-classes)

*

[Further

Classes](../../docs/package_reference/sentence_transformer/models.html#further-classes)

* [quantization](../../docs/package_reference/sentence_transformer/quantization.html)

*

[`quantize_embeddings()`](../../docs/package_reference/sentence_transformer/quantization.html#sentence-transformers.quantization.quantize_embeddings)

*

[`semantic_search_faiss()`](../../docs/package_reference/sentence_transformer/quantization.html#sentence-transformers.quantization.semantic_search_faiss)

*

[`semantic_search_usearch()`](../../docs/package_reference/sentence_transformer/quantization.html#sentence-transformers.quantization.semantic_search_usearch)

* [Cross Encoder](../../docs/package_reference/cross_encoder/index.html)

* [CrossEncoder](../../docs/package_reference/cross_encoder/cross_encoder.html)

- * [CrossEncoder](../../docs/package_reference/cross_encoder/cross_encoder.html#id1)
- * [Training Inputs](../../docs/package_reference/cross_encoder/cross_encoder.html#training-inputs)
- * [Evaluation](../../docs/package_reference/cross_encoder/evaluation.html)
- * [CEBinaryAccuracyEvaluator](../../docs/package_reference/cross_encoder/evaluation.html#cebinaryaccuracyevaluator)
- * [CEBinaryClassificationEvaluator](../../docs/package_reference/cross_encoder/evaluation.html#cebinaryclassificationevaluator)
- * [CECorrelationEvaluator](../../docs/package_reference/cross_encoder/evaluation.html#cecorrelationevaluator)
- * [CEF1Evaluator](../../docs/package_reference/cross_encoder/evaluation.html#cef1evaluator)
- * [CESoftmaxAccuracyEvaluator](../../docs/package_reference/cross_encoder/evaluation.html#cesoftmaxaccuracyevaluator)
- * [CERerankingEvaluator](../../docs/package_reference/cross_encoder/evaluation.html#cererankingevaluator)
- * [util](../../docs/package_reference/util.html)
- * [Helper Functions](../../docs/package_reference/util.html#module-sentence_transformers.util)
- * [community_detection()](../../docs/package_reference/util.html#sentence_transformers.util.community_detection)
- * [http_get()](../../docs/package_reference/util.html#sentence_transformers.util.http_get)

[`is_training_available()`](../../docs/package_reference/util.html#sentence_transformers.util.is_training_available)

*

[`mine_hard_negatives()`](../../docs/package_reference/util.html#sentence_transformers.util.mine_hard_negatives)

*

[`normalize_embeddings()`](../../docs/package_reference/util.html#sentence_transformers.util.normalize_embeddings)

*

[`paraphrase_mining()`](../../docs/package_reference/util.html#sentence_transformers.util.paraphrase_mining)

*

[`semantic_search()`](../../docs/package_reference/util.html#sentence_transformers.util.semantic_search)

*

[`truncate_embeddings()`](../../docs/package_reference/util.html#sentence_transformers.util.truncate_embeddings)

*

[Model

Optimization](../../docs/package_reference/util.html#module-sentence_transformers.backend)

*

[`export_dynamic_quantized_onnx_model()`](../../docs/package_reference/util.html#sentence_transformers.backend.export_dynamic_quantized_onnx_model)

*

[`export_optimized_onnx_model()`](../../docs/package_reference/util.html#sentence_transformers.backend.export_optimized_onnx_model)

*

[`export_static_quantized_openvino_model()`](../../docs/package_reference/util.html#sentence_tra

nsformers.backend.export_static_quantized_openvino_model)

* [Similarity Metrics](../../docs/package_reference/util.html#module-sentence_transformers.util)

* [cos_sim()](../../docs/package_reference/util.html#sentence_transformers.util.cos_sim)

* [dot_score()](../../docs/package_reference/util.html#sentence_transformers.util.dot_score)

*

[euclidean_sim()](../../docs/package_reference/util.html#sentence_transformers.util.euclidean_sim)

*

[manhattan_sim()](../../docs/package_reference/util.html#sentence_transformers.util.manhattan_sim)

*

[pairwise_cos_sim()](../../docs/package_reference/util.html#sentence_transformers.util.pairwise_cos_sim)

*

[pairwise_dot_score()](../../docs/package_reference/util.html#sentence_transformers.util.pairwise_dot_score)

*

[pairwise_euclidean_sim()](../../docs/package_reference/util.html#sentence_transformers.util.pairwise_euclidean_sim)

*

[pairwise_manhattan_sim()](../../docs/package_reference/util.html#sentence_transformers.util.pairwise_manhattan_sim)

__[Sentence Transformers](../../index.html)

* [(../../index.html)]

* [Training Examples](../../docs/sentence_transformer/training/examples.html)

* Model Distillation

*

[

Edit

on

GitHub](https://github.com/UKPLab/sentence-transformers/blob/master/examples/training/distillation/README.md)

* * *

Model Distillation

This folder contains example to make SentenceTransformer models **faster**, **cheaper** and **lighter**. These light models achieve 97.5% - 100% performance of the original model on downstream tasks.

Knowledge Distillation

Knowledge distillation describes the process to transfer knowledge from a teacher model to a student model. It can be used to extend sentence embeddings to new languages ([Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation](https://arxiv.org/abs/2004.09813)), but the traditional approach is to have a slow (but well performing) teacher model and a fast student model.

The fast student model imitates the teacher model and achieves by this a high performance.

We implement two options for creating the student model:

1.

[model_distillation.py](https://github.com/UKPLab/sentence-transformers/tree/master/examples/training/distillation/model_distillation.py): Use a light transformer model like TinyBERT or BERT-Small to imitate the bigger teacher.

2.

[model_distillation_layer_reduction.py](https://github.com/UKPLab/sentence-transformers/tree/master/examples/training/distillation/model_distillation_layer_reduction.py): We take the teacher model and keep only certain layers, for example, only 4 layers.

Option 2) works usually better, as we keep most of the weights from the teacher. In Option 1, we have to tune all weights in the student from scratch.

Speed - Performance Trade-Off

Smaller models are faster, but show a (slightly) worse performance when evaluated on down stream tasks. To get an impression of this trade-off, we show some numbers of the [sts-b-roberta-base](https://huggingface.co/sentence-transformers/stsb-roberta-base) model with different number of layers:

Layers	STSbenchmark Performance	Performance Decrease	Speed (Sent. / Sec. on V100-GPU)
--------	--------------------------	----------------------	----------------------------------

---	---	---	---
-----	-----	-----	-----

teacher: 12	85.44	-	2300
-------------	-------	---	------

8 | 85.54 | +0.1% | 3200 (~1.4x)

6 | 85.23 | -0.2% | 4000 (~1.7x)

4 | 84.92 | -0.6% | 5300 (~2.3x)

3 | 84.39 | -1.2% | 6500 (~2.8x)

2 | 83.32 | -2.5% | 7700 (~3.3x)

1 | 80.86 | -5.4% | 9200 (~4.0x)

Dimensionality Reduction

Warning

Since writing this, [Embedding Quantization](../applications/embedding-quantization/README.html) has been introduced as the go-to approach for shrinking embedding sizes. Following [Thakur et al.](https://arxiv.org/abs/2205.11498), We recommend that approach over PCA.

By default, the pretrained models output embeddings with size 768 (base-models) or with size 1024 (large-models). However, when you store millions of embeddings, this can require quite a lot of memory / storage.

[dimensionality_reduction.py](https://github.com/UKPLab/sentence-transformers/tree/master/examples/training/distillation/dimensionality_reduction.py) contains a simple example how to reduce the embedding dimension to any size by using Principle Component Analysis (PCA). In that example, we reduce 768 dimension to 128 dimension, reducing the storage requirement by factor 6. The performance only slightly drops from 85.44 to 84.96 on the STS benchmark dataset.

This dimensionality reduction technique can easily be applied to existent models. We could even reduce the embeddings size to 32, reducing the storage requirement by factor 24 (performance decreases to 81.82).

Note: This technique neither improves the runtime, nor the memory requirement for running the model. It only reduces the needed space to store embeddings, for example, for [semantic search](../../applications/semantic-search/README.html).

Quantization

A [quantized model](https://pytorch.org/docs/stable/quantization.html) executes some or all of the operations with integers rather than floating point values. This allows for a more compact models and the use of high performance vectorized operations on many hardware platforms.

For models that are run on **CPUs** , this can yield 40% smaller models and a faster inference time: Depending on the CPU, speedup are between 15% and 400%. Model quantization is (as of now) not supported for GPUs by PyTorch.

For an example, see

[model_quantization.py](https://github.com/UKPLab/sentence-transformers/tree/master/examples/training/distillation/model_quantization.py)

Note

The quantization support of Sentence Transformers is still being improved.

[Previous](../multilingual/README.html "Multilingual Models") [Next
(../data_augmentation/README.html "Augmented SBERT")

* * *

(C) Copyright 2025.

Built with [Sphinx](https://www.sphinx-doc.org/) using a
[theme](https://github.com/readthedocs/sphinx_rtd_theme) provided by [Read the
Docs](https://readthedocs.org).