

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

Getting Started

- * [Installation](../..../installation.html)
- * [Install with pip](../..../installation.html#install-with-pip)
- * [Install with Conda](../..../installation.html#install-with-conda)
- * [Install from Source](../..../installation.html#install-from-source)
- * [Editable Install](../..../installation.html#editable-install)
- * [Install PyTorch with CUDA support](../..../installation.html#install-pytorch-with-cuda-support)
- * [Quickstart](../..../quickstart.html)
- * [Sentence Transformer](../..../quickstart.html#sentence-transformer)
- * [Cross Encoder](../..../quickstart.html#cross-encoder)
- * [Next Steps](../..../quickstart.html#next-steps)

Sentence Transformer

- * [Usage](../..../sentence_transformer/usage/usage.html)
- * [Computing Embeddings](../..../examples/applications/computing-embeddings/README.html)
 - * [Initializing a Sentence Transformer Model](../..../examples/applications/computing-embeddings/README.html#initializing-a-sentence-transformer-model)
 - * [Calculating Embeddings](../..../examples/applications/computing-embeddings/README.html#calculating-embeddings)
 - * [Prompt Templates](../..../examples/applications/computing-embeddings/README.html#prompt-templates)

[Length\]\(../../examples/applications/computing-embeddings/README.html#id1\)](#)

[* \[Multi-Process / Multi-GPU Encoding\]\(../../examples/applications/computing-embeddings/README.html#multi-process-multi-gpu-encoding\)](#)

[* \[Semantic Textual Similarity\]\(../../sentence_transformer/usage/semantic_textual_similarity.html\)](#)

[* \[Similarity Calculation\]\(../../sentence_transformer/usage/semantic_textual_similarity.html#similarity-calculation\)](#)

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

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

[* \[Symmetric vs. Asymmetric Semantic Search\]\(../../examples/applications/semantic-search/README.html#symmetric-vs-asymmetric-semantic-search\)](#)

[* \[Manual Implementation\]\(../../examples/applications/semantic-search/README.html#manual-implementation\)](#)

[* \[Optimized Implementation\]\(../../examples/applications/semantic-search/README.html#optimized-implementation\)](#)

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

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

[* \[Approximate Nearest Neighbor\]\(../../examples/applications/semantic-search/README.html#approximate-nearest-neighbor\)](#)

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

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

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

* [Retrieve & Re-Rank

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

* [Retrieval:

Bi-Encoder](../../examples/applications/retrieve_rerank/README.html#retrieval-bi-encoder)

* [Re-Ranker:

Cross-Encoder](../../examples/applications/retrieve_rerank/README.html#re-ranker-cross-encoder)

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

* [Pre-trained Bi-Encoders

(Retrieval)](../../examples/applications/retrieve_rerank/README.html#pre-trained-bi-encoders-retrieval)

* [Pre-trained Cross-Encoders

(Re-Ranker)](../../examples/applications/retrieve_rerank/README.html#pre-trained-cross-encoders-re-ranker)

* [Clustering](../../examples/applications/clustering/README.html)

* [k-Means](../../examples/applications/clustering/README.html#k-means)

* [Agglomerative

Clustering](../../examples/applications/clustering/README.html#agglomerative-clustering)

* [Fast Clustering](../../examples/applications/clustering/README.html#fast-clustering)

* [Topic Modeling](../../examples/applications/clustering/README.html#topic-modeling)

* [Paraphrase Mining](../../examples/applications/paraphrase-mining/README.html)

*

[`paraphrase_mining()`](../../examples/applications/paraphrase-mining/README.html#sentence_transformers.util.paraphrase_mining)

* [Translated Sentence

Mining](../../examples/applications/parallel-sentence-mining/README.html)

* [Margin Based

Mining](../../examples/applications/parallel-sentence-mining/README.html#margin-based-mining)

* [Examples](../../examples/applications/parallel-sentence-mining/README.html#examples)

* [Image Search](../../examples/applications/image-search/README.html)

* [Installation](../../examples/applications/image-search/README.html#installation)

* [Usage](../../examples/applications/image-search/README.html#usage)

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

* [Embedding Quantization](../../examples/applications/embedding-quantization/README.html)

* [Binary

Quantization](../../examples/applications/embedding-quantization/README.html#binary-quantization)

* [Scalar (int8)

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

* [Additional

extensions](../../examples/applications/embedding-quantization/README.html#additional-extensions)

* [Demo](../../examples/applications/embedding-quantization/README.html#demo)

* [Try it

yourself](../../examples/applications/embedding-quantization/README.html#try-it-yourself)

* [Speeding up Inference](../../sentence_transformer/usage/efficiency.html)

* [PyTorch](../../sentence_transformer/usage/efficiency.html#pytorch)

* [ONNX](../../sentence_transformer/usage/efficiency.html#onnx)

* [OpenVINO](../../sentence_transformer/usage/efficiency.html#openvino)

* [Benchmarks](../../sentence_transformer/usage/efficiency.html#benchmarks)

* [Creating Custom Models](../../sentence_transformer/usage/custom_models.html)

* [Structure of Sentence Transformer

Models](../../sentence_transformer/usage/custom_models.html#structure-of-sentence-transformer-models)

* [Sentence Transformer Model from a Transformers

Model](../../sentence_transformer/usage/custom_models.html#sentence-transformer-model-from-a-transformers-model)

* [Pretrained Models](../../sentence_transformer/pretrained_models.html)

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

* [Semantic Search

Models](../../sentence_transformer/pretrained_models.html#semantic-search-models)

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

* [MSMARCO Passage

Models](../../sentence_transformer/pretrained_models.html#msmarco-passage-models)

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

* [Semantic Similarity

Models](../../sentence_transformer/pretrained_models.html#semantic-similarity-models)

* [Bitext Mining](../../sentence_transformer/pretrained_models.html#bitext-mining)

* [Image & Text-Models](../../sentence_transformer/pretrained_models.html#image-text-models)

* [INSTRUCTOR models](../../sentence_transformer/pretrained_models.html#instructor-models)

* [Scientific Similarity

Models](../../sentence_transformer/pretrained_models.html#scientific-similarity-models)

* [Training Overview](../../sentence_transformer/training_overview.html)

* [Why Finetune?](../../sentence_transformer/training_overview.html#why-finetune)

* [Training Components](../../sentence_transformer/training_overview.html#training-components)

* [Dataset](../../sentence_transformer/training_overview.html#dataset)

* [Dataset Format](../../sentence_transformer/training_overview.html#dataset-format)

* [Loss Function](../../sentence_transformer/training_overview.html#loss-function)

- * [Training Arguments](../../sentence_transformer/training_overview.html#training-arguments)
- * [Evaluator](../../sentence_transformer/training_overview.html#evaluator)
- * [Trainer](../../sentence_transformer/training_overview.html#trainer)
- * [Callbacks](../../sentence_transformer/training_overview.html#callbacks)
- * [Multi-Dataset Training](../../sentence_transformer/training_overview.html#multi-dataset-training)
- * [Deprecated Training](../../sentence_transformer/training_overview.html#deprecated-training)
- * [Best Base Embedding Models](../../sentence_transformer/training_overview.html#best-base-embedding-models)
- * [Dataset Overview](../../sentence_transformer/dataset_overview.html)
 - * [Datasets on the Hugging Face Hub](../../sentence_transformer/dataset_overview.html#datasets-on-the-hugging-face-hub)
 - * [Pre-existing Datasets](../../sentence_transformer/dataset_overview.html#pre-existing-datasets)
- * [Loss Overview](../../sentence_transformer/loss_overview.html)
 - * [Loss modifiers](../../sentence_transformer/loss_overview.html#loss-modifiers)
 - * [Distillation](../../sentence_transformer/loss_overview.html#distillation)
 - * [Commonly used Loss Functions](../../sentence_transformer/loss_overview.html#commonly-used-loss-functions)
 - * [Custom Loss Functions](../../sentence_transformer/loss_overview.html#custom-loss-functions)
- * [Training Examples](../../sentence_transformer/training/examples.html)
 - * [Semantic Textual Similarity](../../examples/training/sts/README.html)
 - * [Training data](../../examples/training/sts/README.html#training-data)
 - * [Loss Function](../../examples/training/sts/README.html#loss-function)
 - * [Natural Language Inference](../../examples/training/nli/README.html)
 - * [Data](../../examples/training/nli/README.html#data)
 - * [SoftmaxLoss](../../examples/training/nli/README.html#softmaxloss)
 - * [MultipleNegativesRankingLoss](../../examples/training/nli/README.html#multiplenegativesrankin

gloss)

- * [Paraphrase Data](../../examples/training/paraphrases/README.html)
- * [Pre-Trained Models](../../examples/training/paraphrases/README.html#pre-trained-models)
- * [Quora Duplicate Questions](../../examples/training/quora_duplicate_questions/README.html)
- * [Training](../../examples/training/quora_duplicate_questions/README.html#training)

*

[MultipleNegativesRankingLoss](../../examples/training/quora_duplicate_questions/README.html#multiplenegativesrankingloss)

*

[Pretrained

Models](../../examples/training/quora_duplicate_questions/README.html#pretrained-models)

- * [MS MARCO](../../examples/training/ms_marco/README.html)
- * [Bi-Encoder](../../examples/training/ms_marco/README.html#bi-encoder)
- * [Matryoshka Embeddings](../../examples/training/matryoshka/README.html)
- * [Use Cases](../../examples/training/matryoshka/README.html#use-cases)
- * [Results](../../examples/training/matryoshka/README.html#results)
- * [Training](../../examples/training/matryoshka/README.html#training)
- * [Inference](../../examples/training/matryoshka/README.html#inference)
- * [Code Examples](../../examples/training/matryoshka/README.html#code-examples)
- * [Adaptive Layers](../../examples/training/adaptive_layer/README.html)
- * [Use Cases](../../examples/training/adaptive_layer/README.html#use-cases)
- * [Results](../../examples/training/adaptive_layer/README.html#results)
- * [Training](../../examples/training/adaptive_layer/README.html#training)
- * [Inference](../../examples/training/adaptive_layer/README.html#inference)
- * [Code Examples](../../examples/training/adaptive_layer/README.html#code-examples)
- * [Multilingual Models](../../examples/training/multilingual/README.html)

*

[Extend

your

own

models](../../examples/training/multilingual/README.html#extend-your-own-models)

- * [Training](../../examples/training/multilingual/README.html#training)
- * [Datasets](../../examples/training/multilingual/README.html#datasets)
 - * [Sources for Training Data](../../examples/training/multilingual/README.html#sources-for-training-data)
 - * [Evaluation](../../examples/training/multilingual/README.html#evaluation)
 - * [Available Pre-trained Models](../../examples/training/multilingual/README.html#available-pre-trained-models)
 - * [Usage](../../examples/training/multilingual/README.html#usage)
 - * [Performance](../../examples/training/multilingual/README.html#performance)
 - * [Citation](../../examples/training/multilingual/README.html#citation)
 - * [Model Distillation](../../examples/training/distillation/README.html)
 - * [Knowledge Distillation](../../examples/training/distillation/README.html#knowledge-distillation)
 - * [Speed - Performance Trade-Off](../../examples/training/distillation/README.html#speed-performance-trade-off)
 - * [Dimensionality Reduction](../../examples/training/distillation/README.html#dimensionality-reduction)
 - * [Quantization](../../examples/training/distillation/README.html#quantization)
 - * [Augmented SBERT](../../examples/training/data_augmentation/README.html)
 - * [Motivation](../../examples/training/data_augmentation/README.html#motivation)
 - * [Extend to your own datasets](../../examples/training/data_augmentation/README.html#extend-to-your-own-datasets)
 - * [Methodology](../../examples/training/data_augmentation/README.html#methodology)
 - * [Scenario 1: Limited or small annotated datasets (few labeled sentence-pairs)](../../examples/training/data_augmentation/README.html#scenario-1-limited-or-small-annotated-datasets-few-labeled-sentence-pairs)
 - * [Scenario 2: No annotated datasets (Only unlabeled

[sentence-pairs\)\]\(../../examples/training/data_augmentation/README.html#scenario-2-no-annotated-datasets-only-unlabeled-sentence-pairs\)](#)

- * [\[Training\]\(../../examples/training/data_augmentation/README.html#training\)](#)
- * [\[Citation\]\(../../examples/training/data_augmentation/README.html#citation\)](#)
- * [\[Training with Prompts\]\(../../examples/training/prompts/README.html\)](#)
- * [\[What are Prompts?\]\(../../examples/training/prompts/README.html#what-are-prompts\)](#)
 - * [\[Why would we train with Prompts?\]\(../../examples/training/prompts/README.html#why-would-we-train-with-prompts\)](#)
 - * [\[How do we train with Prompts?\]\(../../examples/training/prompts/README.html#how-do-we-train-with-prompts\)](#)
- * [\[Training with PEFT Adapters\]\(../../examples/training/peft/README.html\)](#)
- * [\[Compatibility Methods\]\(../../examples/training/peft/README.html#compatibility-methods\)](#)
- * [\[Adding a New Adapter\]\(../../examples/training/peft/README.html#adding-a-new-adapter\)](#)
 - * [\[Loading a Pretrained Adapter\]\(../../examples/training/peft/README.html#loading-a-pretrained-adapter\)](#)
- * [\[Training Script\]\(../../examples/training/peft/README.html#training-script\)](#)
- * [\[Unsupervised Learning\]\(../../examples/unsupervised_learning/README.html\)](#)
- * [\[TSDAE\]\(../../examples/unsupervised_learning/README.html#tsdae\)](#)
- * [\[SimCSE\]\(../../examples/unsupervised_learning/README.html#simcse\)](#)
- * [\[CT\]\(../../examples/unsupervised_learning/README.html#ct\)](#)
 - * [\[CT \(In-Batch Negative Sampling\)\]\(../../examples/unsupervised_learning/README.html#ct-in-batch-negative-sampling\)](#)
 - * [\[Masked Language Model \(MLM\)\]\(../../examples/unsupervised_learning/README.html#masked-language-model-mlm\)](#)
- * [\[GenQ\]\(../../examples/unsupervised_learning/README.html#genq\)](#)
- * [\[GPL\]\(../../examples/unsupervised_learning/README.html#gpl\)](#)
- * [\[Performance](#)

[Comparison\]\(../../examples/unsupervised_learning/README.html#performance-comparison\)](#)

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

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

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

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

* [\[Hyperparameter Optimization\]\(../../examples/training/hpo/README.html\)](#)

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

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

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

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

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

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

Cross Encoder

* [\[Usage\]\(../../cross_encoder/usage/usage.html\)](#)

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

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

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

* [\[Re-Ranker:](#)

Cross-Encoder](../../../../examples/applications/retrieve_rerank/README.html#re-ranker-cross-encoder)

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

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

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

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

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

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

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

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

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

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

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

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

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

*

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

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

Package Reference

* Sentence Transformer

* [SentenceTransformer](SentenceTransformer.html)

* [SentenceTransformer](SentenceTransformer.html#id1)

*

[SentenceTransformerModelCardData](SentenceTransformer.html#sentencetransformermodelcarddata)

* [SimilarityFunction](SentenceTransformer.html#similarityfunction)

* [Trainer](trainer.html)

* [SentenceTransformerTrainer](trainer.html#sentencetransformertrainer)

* [Training Arguments](training_args.html)

*

[SentenceTransformerTrainingArguments](training_args.html#sentencetransformertrainingarguments)

* [Losses](losses.html)

* [BatchAllTripletLoss](losses.html#batchalltripletloss)

* [BatchHardSoftMarginTripletLoss](losses.html#batchhardsoftmargintripletloss)

* [BatchHardTripletLoss](losses.html#batchhardtripletloss)

* [BatchSemiHardTripletLoss](losses.html#batchsemihardtripletloss)

* [ContrastiveLoss](losses.html#contrastiveloss)

* [OnlineContrastiveLoss](losses.html#onlinecontrastiveloss)

* [ContrastiveTensionLoss](losses.html#contrastivetensionloss)

*

[ContrastiveTensionLossInBatchNegatives](losses.html#contrastivetensionlossinbatchnegatives)

* [CoSENTLoss](losses.html#cosentloss)

* [AngleELoss](losses.html#angleloss)

* [CosineSimilarityLoss](losses.html#cosinesimilarityloss)

- * [DenoisingAutoEncoderLoss](losses.html#denoisingautoencoderloss)
- * [GISTEmbedLoss](losses.html#gistembedloss)
- * [CachedGISTEmbedLoss](losses.html#cachedgistembedloss)
- * [MSELoss](losses.html#mseloss)
- * [MarginMSELoss](losses.html#marginmseloss)
- * [MatryoshkaLoss](losses.html#matryoshkaloss)
- * [Matryoshka2dLoss](losses.html#matryoshka2dloss)
- * [AdaptiveLayerLoss](losses.html#adaptivelayerloss)
- * [MegaBatchMarginLoss](losses.html#megabatchmarginloss)
- * [MultipleNegativesRankingLoss](losses.html#multiplenegativesrankingloss)
- * [CachedMultipleNegativesRankingLoss](losses.html#cachedmultiplenegativesrankingloss)

*

[MultipleNegativesSymmetricRankingLoss](losses.html#multiplenegativessymmetricrankingloss)

*

[CachedMultipleNegativesSymmetricRankingLoss](losses.html#cachedmultiplenegativessymmetricrankingloss)

- * [SoftmaxLoss](losses.html#softmaxloss)
- * [TripletLoss](losses.html#tripletloss)
- * [Samplers](sampler.html)
- * [BatchSamplers](sampler.html#batchsamplers)
- * [MultiDatasetBatchSamplers](sampler.html#multidatasetbatchsamplers)
- * [Evaluation](evaluation.html)
- * [BinaryClassificationEvaluator](evaluation.html#binaryclassificationevaluator)
- * [EmbeddingSimilarityEvaluator](evaluation.html#embeddingsimilarityevaluator)
- * [InformationRetrievalEvaluator](evaluation.html#informationretrievalevaluator)
- * [NanoBEIREvaluator](evaluation.html#nanobeirevaluator)
- * [MSEEvaluator](evaluation.html#mseevaluator)

- * [ParaphraseMiningEvaluator](evaluation.html#paraphraseminingevaluator)
- * [RerankingEvaluator](evaluation.html#rerankingevaluator)
- * [SentenceEvaluator](evaluation.html#sentenceevaluator)
- * [SequentialEvaluator](evaluation.html#sequentialevaluator)
- * [TranslationEvaluator](evaluation.html#translationevaluator)
- * [TripletEvaluator](evaluation.html#tripletevaluator)
- * [Datasets](datasets.html)
- * [ParallelSentencesDataset](datasets.html#parallelsentencesdataset)
- * [SentenceLabelDataset](datasets.html#sentencelabeldataset)
- * [DenoisingAutoEncoderDataset](datasets.html#denoisingautoencoderdataset)
- * [NoDuplicatesDataLoader](datasets.html#noduplicatesdataloader)
- * [Models](models.html)
- * [Main Classes](models.html#main-classes)
- * [Further Classes](models.html#further-classes)
- * [quantization](quantization.html)

*

[`quantize_embeddings()`](quantization.html#sentence_transformers.quantization.quantize_embeddings)

*

[`semantic_search_faiss()`](quantization.html#sentence_transformers.quantization.semantic_search_faiss)

*

[`semantic_search_usearch()`](quantization.html#sentence_transformers.quantization.semantic_search_usearch)

- * [Cross Encoder](../cross_encoder/index.html)
- * [CrossEncoder](../cross_encoder/cross_encoder.html)
- * [CrossEncoder](../cross_encoder/cross_encoder.html#id1)

- * [Training Inputs](../cross_encoder/cross_encoder.html#training-inputs)
- * [Evaluation](../cross_encoder/evaluation.html)
- * [CEBinaryAccuracyEvaluator](../cross_encoder/evaluation.html#cebinaryaccuracyevaluator)

*

[CEBinaryClassificationEvaluator](../cross_encoder/evaluation.html#cebinaryclassificationevaluator)

- * [CECorrelationEvaluator](../cross_encoder/evaluation.html#cecorrelationevaluator)
- * [CEF1Evaluator](../cross_encoder/evaluation.html#cef1evaluator)

*

[CESoftmaxAccuracyEvaluator](../cross_encoder/evaluation.html#cesoftmaxaccuracyevaluator)

- * [CERerankingEvaluator](../cross_encoder/evaluation.html#cererankingevaluator)

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

* [Helper Functions](../util.html#module-sentence_transformers.util)

* [community_detection()](../util.html#sentence_transformers.util.community_detection)

* [http_get()](../util.html#sentence_transformers.util.http_get)

* [is_training_available()](../util.html#sentence_transformers.util.is_training_available)

* [mine_hard_negatives()](../util.html#sentence_transformers.util.mine_hard_negatives)

* [normalize_embeddings()](../util.html#sentence_transformers.util.normalize_embeddings)

* [paraphrase_mining()](../util.html#sentence_transformers.util.paraphrase_mining)

* [semantic_search()](../util.html#sentence_transformers.util.semantic_search)

* [truncate_embeddings()](../util.html#sentence_transformers.util.truncate_embeddings)

* [Model Optimization](../util.html#module-sentence_transformers.backend)

*

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

*

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

[`export_static_quantized_openvino_model()](../util.html#sentence_transformers.backend.export_static_quantized_openvino_model)

- * [Similarity Metrics](../util.html#module-sentence_transformers.util)
- * [cos_sim()](../util.html#sentence_transformers.util.cos_sim)
- * [dot_score()](../util.html#sentence_transformers.util.dot_score)
- * [euclidean_sim()](../util.html#sentence_transformers.util.euclidean_sim)
- * [manhattan_sim()](../util.html#sentence_transformers.util.manhattan_sim)
- * [pairwise_cos_sim()](../util.html#sentence_transformers.util.pairwise_cos_sim)
- * [pairwise_dot_score()](../util.html#sentence_transformers.util.pairwise_dot_score)
- * [pairwise_euclidean_sim()](../util.html#sentence_transformers.util.pairwise_euclidean_sim)
- * [pairwise_manhattan_sim()](../util.html#sentence_transformers.util.pairwise_manhattan_sim)

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

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

* Sentence Transformer

* [Edit on

GitHub](https://github.com/UKPLab/sentence-transformers/blob/master/docs/package_reference/sentence_transformer/index.rst)

* * *

Sentence Transformer *if*•

- * [SentenceTransformer](SentenceTransformer.html)
- * [SentenceTransformer](SentenceTransformer.html#id1)

*

[`SentenceTransformer`](SentenceTransformer.html#sentence_transformers.SentenceTransformer)

*

[`SentenceTransformer.active_adapters()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.active_adapters)

*

[`SentenceTransformer.add_adapter()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.add_adapter)

*

[`SentenceTransformer.bfloat16()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.bfloat16)

*

[`SentenceTransformer.compile()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.compile)

*

[`SentenceTransformer.cpu()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.cpu)

*

[`SentenceTransformer.cuda()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.cuda)

*

[`SentenceTransformer.device`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.device)

*

[`SentenceTransformer.disable_adapters()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.disable_adapters)

*

[`SentenceTransformer.double()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.double)

*

[`SentenceTransformer.enable_adapters()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.enable_adapters)

*

[`SentenceTransformer.encode()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.encode)

*

[`SentenceTransformer.encode_multi_process()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.encode_multi_process)

*

[`SentenceTransformer.eval()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.eval)

*

[`SentenceTransformer.evaluate()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.evaluate)

*

[`SentenceTransformer.fit()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.fit)

*

[`SentenceTransformer.float()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.float)

*

[`SentenceTransformer.get_adapter_state_dict()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.get_adapter_state_dict)

*

[`SentenceTransformer.get_backend()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.get_backend)

*

[`SentenceTransformer.get_max_seq_length()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.get_max_seq_length)

*

[`SentenceTransformer.get_sentence_embedding_dimension()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.get_sentence_embedding_dimension)

*

[`SentenceTransformer.half()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.half)

*

[`SentenceTransformer.load_adapter()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.load_adapter)

*

[`SentenceTransformer.max_seq_length`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.max_seq_length)

*

[`SentenceTransformer.old_fit()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.old_fit)

*

[`SentenceTransformer.push_to_hub()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.push_to_hub)

*

[`SentenceTransformer.save()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.save)

*

[`SentenceTransformer.save_pretrained()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.save_pretrained)

*

[`SentenceTransformer.set_adapter()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.set_adapter)

*

[`SentenceTransformer.set_pooling_include_prompt()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.set_pooling_include_prompt)

*

[`SentenceTransformer.similarity`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.similarity)

*

[`SentenceTransformer.similarity_fn_name`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.similarity_fn_name)

*

[`SentenceTransformer.similarity_pairwise`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.similarity_pairwise)

*

[`SentenceTransformer.smart_batching_collate()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.smart_batching_collate)

*

[`SentenceTransformer.start_multi_process_pool()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.start_multi_process_pool)

*

[`SentenceTransformer.stop_multi_process_pool()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.stop_multi_process_pool)

*

[`SentenceTransformer.to()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.to)

*

[`SentenceTransformer.tokenize()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.tokenize)

*

[`SentenceTransformer.tokenizer`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.tokenizer)

*

[`SentenceTransformer.train()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.train)

*

[`SentenceTransformer.truncate_sentence_embeddings()`](SentenceTransformer.html#sentence_transformers.SentenceTransformer.truncate_sentence_embeddings)

*

[SentenceTransformerModelCardData](SentenceTransformer.html#sentence_transformers.model_card.SentenceTransformerModelCardData)

*

[`SentenceTransformerModelCardData`](SentenceTransformer.html#sentence_transformers.model_card.SentenceTransformerModelCardData)

* [SimilarityFunction](SentenceTransformer.html#similarityfunction)

* [`SimilarityFunction`](SentenceTransformer.html#sentence_transformers.SimilarityFunction)

*

[`SimilarityFunction.possible_values()`](SentenceTransformer.html#sentence_transformers.SimilarityFunction.possible_values)

*

[`SimilarityFunction.to_similarity_fn()`](SentenceTransformer.html#sentence_transformers.SimilarityFunction.to_similarity_fn)

Function.to_similarity_fn)

*

[`SimilarityFunction.to_similarity_pairwise_fn()`](SentenceTransformer.html#sentence_transformers.

SimilarityFunction.to_similarity_pairwise_fn)

* [Trainer](trainer.html)

* [SentenceTransformerTrainer](trainer.html#sentence_transformertrainer)

*

[`SentenceTransformerTrainer`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer)

*

[`SentenceTransformerTrainer.add_callback()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.add_callback)

*

[`SentenceTransformerTrainer.add_model_card_callback()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.add_model_card_callback)

*

[`SentenceTransformerTrainer.add_prompts_or_dataset_name_transform()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.add_prompts_or_dataset_name_transform)

*

[`SentenceTransformerTrainer.compute_loss()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.compute_loss)

*

[`SentenceTransformerTrainer.create_model_card()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.create_model_card)

*

[`SentenceTransformerTrainer.create_optimizer()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.create_optimizer)

*

[`SentenceTransformerTrainer.create_optimizer_and_scheduler()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.create_optimizer_and_scheduler)

*

[`SentenceTransformerTrainer.create_scheduler()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.create_scheduler)

*

[`SentenceTransformerTrainer.evaluate()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.evaluate)

*

[`SentenceTransformerTrainer.get_batch_sampler()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.get_batch_sampler)

*

[`SentenceTransformerTrainer.get_eval_dataloader()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.get_eval_dataloader)

*

[`SentenceTransformerTrainer.get_learning_rates()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.get_learning_rates)

*

[`SentenceTransformerTrainer.get_multi_dataset_batch_sampler()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.get_multi_dataset_batch_sampler)

*

[`SentenceTransformerTrainer.get_num_trainable_parameters()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.get_num_trainable_parameters)

*

[`SentenceTransformerTrainer.get_optimizer_group()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.get_optimizer_group)

*

[`SentenceTransformerTrainer.get_test_dataloader()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.get_test_dataloader)

*

[`SentenceTransformerTrainer.get_train_dataloader()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.get_train_dataloader)

*

[`SentenceTransformerTrainer.hyperparameter_search()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.hyperparameter_search)

*

[`SentenceTransformerTrainer.is_local_process_zero()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.is_local_process_zero)

*

[`SentenceTransformerTrainer.is_world_process_zero()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.is_world_process_zero)

*

[`SentenceTransformerTrainer.log()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.log)

*

[`SentenceTransformerTrainer.maybe_add_prompts_or_dataset_name_column()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.maybe_add_prompts_or_dataset_name_column)

*

[`SentenceTransformerTrainer.pop_callback()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.pop_callback)

*

[`SentenceTransformerTrainer.propagate_args_to_deepspeed()`](trainer.html#sentence_transforme

rs.trainer.SentenceTransformerTrainer.propagate_args_to_deepspeed)

*

[`SentenceTransformerTrainer.push_to_hub()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.push_to_hub)

*

[`SentenceTransformerTrainer.remove_callback()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.remove_callback)

*

[`SentenceTransformerTrainer.save_model()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.save_model)

*

[`SentenceTransformerTrainer.train()`](trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.train)

* [Training Arguments](training_args.html)

*

[SentenceTransformerTrainingArguments](training_args.html#sentence_transformers.training_arguments)

*

[`SentenceTransformerTrainingArguments`](training_args.html#sentence_transformers.training_arguments.SentenceTransformerTrainingArguments)

*

[`SentenceTransformerTrainingArguments.ddp_timeout_delta`](training_args.html#sentence_transformers.training_arguments.SentenceTransformerTrainingArguments.ddp_timeout_delta)

*

[`SentenceTransformerTrainingArguments.device`](training_args.html#sentence_transformers.training_arguments.SentenceTransformerTrainingArguments.device)

*

[`SentenceTransformerTrainingArguments.eval_batch_size`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.eval_batch_size)

*

[`SentenceTransformerTrainingArguments.get_process_log_level`()](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.get_process_log_level)

*

[`SentenceTransformerTrainingArguments.get_warmup_steps`()](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.get_warmup_steps)

*

[`SentenceTransformerTrainingArguments.local_process_index`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.local_process_index)

*

[`SentenceTransformerTrainingArguments.main_process_first`()](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.main_process_first)

*

[`SentenceTransformerTrainingArguments.n_gpu`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.n_gpu)

*

[`SentenceTransformerTrainingArguments.parallel_mode`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.parallel_mode)

*

[`SentenceTransformerTrainingArguments.place_model_on_device`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.place_model_on_device)

*

[`SentenceTransformerTrainingArguments.process_index`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.process_index)

*

[`SentenceTransformerTrainingArguments.set_dataloader()`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.set_dataloader)

*

[`SentenceTransformerTrainingArguments.set_evaluate()`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.set_evaluate)

*

[`SentenceTransformerTrainingArguments.set_logging()`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.set_logging)

*

[`SentenceTransformerTrainingArguments.set_lr_scheduler()`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.set_lr_scheduler)

*

[`SentenceTransformerTrainingArguments.set_optimizer()`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.set_optimizer)

*

[`SentenceTransformerTrainingArguments.set_push_to_hub()`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.set_push_to_hub)

*

[`SentenceTransformerTrainingArguments.set_save()`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.set_save)

*

[`SentenceTransformerTrainingArguments.set_testing()`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.set_testing)

*

[`SentenceTransformerTrainingArguments.set_training()`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.set_training)

*

[`SentenceTransformerTrainingArguments.should_log`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.should_log)

*

[`SentenceTransformerTrainingArguments.should_save`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.should_save)

*

[`SentenceTransformerTrainingArguments.to_dict()`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.to_dict)

*

[`SentenceTransformerTrainingArguments.to_json_string()`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.to_json_string)

*

[`SentenceTransformerTrainingArguments.to_sanitized_dict()`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.to_sanitized_dict)

*

[`SentenceTransformerTrainingArguments.train_batch_size`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.train_batch_size)

*

[`SentenceTransformerTrainingArguments.world_size`](training_args.html#sentence_transformers.training_args.SentenceTransformerTrainingArguments.world_size)

* [Losses](losses.html)

* [BatchAllTripletLoss](losses.html#batchalltripletloss)

* [`BatchAllTripletLoss`](losses.html#sentence_transformers.losses.BatchAllTripletLoss)

* [BatchHardSoftMarginTripletLoss](losses.html#batchhardsoftmargintripletloss)

*

[`BatchHardSoftMarginTripletLoss`](losses.html#sentence_transformers.losses.BatchHardSoftMarginTripletLoss)

* [BatchHardTripletLoss](losses.html#batchhardtripletloss)

* [BatchHardTripletLoss](losses.html#sentence_transformers.losses.BatchHardTripletLoss)

* [BatchSemiHardTripletLoss](losses.html#batchsemihardtripletloss)

*

[BatchSemiHardTripletLoss](losses.html#sentence_transformers.losses.BatchSemiHardTripletLoss)

* [ContrastiveLoss](losses.html#contrastiveloss)

* [ContrastiveLoss](losses.html#sentence_transformers.losses.ContrastiveLoss)

* [OnlineContrastiveLoss](losses.html#onlinecontrastiveloss)

* [OnlineContrastiveLoss](losses.html#sentence_transformers.losses.OnlineContrastiveLoss)

* [ContrastiveTensionLoss](losses.html#contrastivetensionloss)

*

[ContrastiveTensionLoss](losses.html#sentence_transformers.losses.ContrastiveTensionLoss)

*

[ContrastiveTensionLossInBatchNegatives](losses.html#contrastivetensionlossinbatchnegatives)

*

[ContrastiveTensionLossInBatchNegatives](losses.html#sentence_transformers.losses.ContrastiveTensionLossInBatchNegatives)

* [CoSENTLoss](losses.html#cosentloss)

* [CoSENTLoss](losses.html#sentence_transformers.losses.CoSENTLoss)

* [AngleLoss](losses.html#angleloss)

* [AngleLoss](losses.html#sentence_transformers.losses.AngleLoss)

* [CosineSimilarityLoss](losses.html#cosinesimilarityloss)

* [CosineSimilarityLoss](losses.html#sentence_transformers.losses.CosineSimilarityLoss)

* [DenoisingAutoEncoderLoss](losses.html#denoisingautoencoderloss)

*

[DenoisingAutoEncoderLoss](losses.html#sentence_transformers.losses.DenoisingAutoEncoderLoss)

ss)

* [GISTEmbedLoss](losses.html#gistembedloss)

* [`GISTEmbedLoss`](losses.html#sentence_transformers.losses.GISTEmbedLoss)

* [CachedGISTEmbedLoss](losses.html#cachedgistembedloss)

*

[`CachedGISTEmbedLoss`](losses.html#sentence_transformers.losses.CachedGISTEmbedLoss)

* [MSELoss](losses.html#mseloss)

* [`MSELoss`](losses.html#sentence_transformers.losses.MSELoss)

* [MarginMSELoss](losses.html#marginmseloss)

* [`MarginMSELoss`](losses.html#sentence_transformers.losses.MarginMSELoss)

* [MatryoshkaLoss](losses.html#matryoshkaloss)

* [`MatryoshkaLoss`](losses.html#sentence_transformers.losses.MatryoshkaLoss)

* [Matryoshka2dLoss](losses.html#matryoshka2dloss)

* [`Matryoshka2dLoss`](losses.html#sentence_transformers.losses.Matryoshka2dLoss)

* [AdaptiveLayerLoss](losses.html#adaptivelayerloss)

* [`AdaptiveLayerLoss`](losses.html#sentence_transformers.losses.AdaptiveLayerLoss)

* [MegaBatchMarginLoss](losses.html#megabatchmarginloss)

* [`MegaBatchMarginLoss`](losses.html#sentence_transformers.losses.MegaBatchMarginLoss)

* [MultipleNegativesRankingLoss](losses.html#multiplenegativesrankingloss)

*

[`MultipleNegativesRankingLoss`](losses.html#sentence_transformers.losses.MultipleNegativesRankingLoss)

* [CachedMultipleNegativesRankingLoss](losses.html#cachedmultiplenegativesrankingloss)

*

[`CachedMultipleNegativesRankingLoss`](losses.html#sentence_transformers.losses.CachedMultipleNegativesRankingLoss)

* [MultipleNegativesSymmetricRankingLoss](losses.html#multiplenegativessymmetricrankingloss)

*

[`MultipleNegativesSymmetricRankingLoss`](losses.html#sentence_transformers.losses.MultipleNegativesSymmetricRankingLoss)

*

[`CachedMultipleNegativesSymmetricRankingLoss`](losses.html#cachedmultiplenegativessymmetricrankingloss)

*

[`CachedMultipleNegativesSymmetricRankingLoss`](losses.html#sentence_transformers.losses.CachedMultipleNegativesSymmetricRankingLoss)

* [SoftmaxLoss](losses.html#softmaxloss)

* [`SoftmaxLoss`](losses.html#sentence_transformers.losses.SoftmaxLoss)

* [TripletLoss](losses.html#tripletloss)

* [`TripletLoss`](losses.html#sentence_transformers.losses.TripletLoss)

* [Samplers](sampler.html)

* [BatchSamplers](sampler.html#batchsamplers)

* [`BatchSamplers`](sampler.html#sentence_transformers.training_args.BatchSamplers)

* [`DefaultBatchSampler`](sampler.html#sentence_transformers.sampler.DefaultBatchSampler)

*

[`NoDuplicatesBatchSampler`](sampler.html#sentence_transformers.sampler.NoDuplicatesBatchSampler)

*

[`GroupByLabelBatchSampler`](sampler.html#sentence_transformers.sampler.GroupByLabelBatchSampler)

* [MultiDatasetBatchSamplers](sampler.html#multidatasetbatchsamplers)

*

[`MultiDatasetBatchSamplers`](sampler.html#sentence_transformers.training_args.MultiDatasetBatchSamplers)

*

[`RoundRobinBatchSampler`](sampler.html#sentence_transformers.sampler.RoundRobinBatchSampler)

*

[`ProportionalBatchSampler`](sampler.html#sentence_transformers.sampler.ProportionalBatchSampler)

* [Evaluation](evaluation.html)

* [BinaryClassificationEvaluator](evaluation.html#binaryclassificationevaluator)

*

[`BinaryClassificationEvaluator`](evaluation.html#sentence_transformers.evaluation.BinaryClassificationEvaluator)

* [EmbeddingSimilarityEvaluator](evaluation.html#embeddingsimilarityevaluator)

*

[`EmbeddingSimilarityEvaluator`](evaluation.html#sentence_transformers.evaluation.EmbeddingSimilarityEvaluator)

* [InformationRetrievalEvaluator](evaluation.html#informationretrievalevaluator)

*

[`InformationRetrievalEvaluator`](evaluation.html#sentence_transformers.evaluation.InformationRetrievalEvaluator)

* [NanoBEIREvaluator](evaluation.html#nanobeirevaluator)

*

[`NanoBEIREvaluator`](evaluation.html#sentence_transformers.evaluation.NanoBEIREvaluator)

* [MSEEvaluator](evaluation.html#mseevaluator)

* [`MSEEvaluator`](evaluation.html#sentence_transformers.evaluation.MSEEvaluator)

* [ParaphraseMiningEvaluator](evaluation.html#paraphraseminingevaluator)

*

[`ParaphraseMiningEvaluator`](evaluation.html#sentence_transformers.evaluation.ParaphraseMiningEvaluator)

gEvaluator)

- * [RerankingEvaluator](evaluation.html#rerankingevaluator)
- * [RerankingEvaluator](evaluation.html#sentence_transformers.evaluation.RerankingEvaluator)
- * [SentenceEvaluator](evaluation.html#sentenceevaluator)
- * [SentenceEvaluator](evaluation.html#sentence_transformers.evaluation.SentenceEvaluator)
- * [SequentialEvaluator](evaluation.html#sequentialevaluator)

*

[SequentialEvaluator](evaluation.html#sentence_transformers.evaluation.SequentialEvaluator)

- * [TranslationEvaluator](evaluation.html#translationevaluator)

*

[TranslationEvaluator](evaluation.html#sentence_transformers.evaluation.TranslationEvaluator)

- * [TripletEvaluator](evaluation.html#tripletevaluator)
- * [TripletEvaluator](evaluation.html#sentence_transformers.evaluation.TripletEvaluator)

- * [Datasets](datasets.html)

- * [ParallelSentencesDataset](datasets.html#parallelsentencesdataset)

*

[ParallelSentencesDataset](datasets.html#sentence_transformers.datasets.ParallelSentencesDataset)

- * [SentenceLabelDataset](datasets.html#sentencelabeldataset)

*

[SentenceLabelDataset](datasets.html#sentence_transformers.datasets.SentenceLabelDataset)

- * [DenoisingAutoEncoderDataset](datasets.html#denoisingautoencoderdataset)

*

[DenoisingAutoEncoderDataset](datasets.html#sentence_transformers.datasets.DenoisingAutoEncoderDataset)

- * [NoDuplicatesDataLoader](datasets.html#noduplicatesdataloader)

*

[`NoDuplicatesDataLoader`](datasets.html#sentence_transformers.datasets.NoDuplicatesDataLoader)

* [Models](models.html)

* [Main Classes](models.html#main-classes)

* [`Transformer`](models.html#sentence_transformers.models.Transformer)

* [`Pooling`](models.html#sentence_transformers.models.Pooling)

* [`Dense`](models.html#sentence_transformers.models.Dense)

* [Further Classes](models.html#further-classes)

* [`Asym`](models.html#sentence_transformers.models.Asym)

* [`BoW`](models.html#sentence_transformers.models.BoW)

* [`CNN`](models.html#sentence_transformers.models.CNN)

* [`LSTM`](models.html#sentence_transformers.models.LSTM)

* [`Normalize`](models.html#sentence_transformers.models.Normalize)

* [`StaticEmbedding`](models.html#sentence_transformers.models.StaticEmbedding)

*

[`StaticEmbedding.from_distillation()`](models.html#sentence_transformers.models.StaticEmbedding.from_distillation)

*

[`StaticEmbedding.from_model2vec()`](models.html#sentence_transformers.models.StaticEmbedding.from_model2vec)

* [`WeightedLayerPooling`](models.html#sentence_transformers.models.WeightedLayerPooling)

* [`WordEmbeddings`](models.html#sentence_transformers.models.WordEmbeddings)

* [`WordWeights`](models.html#sentence_transformers.models.WordWeights)

* [quantization](quantization.html)

*

[`quantize_embeddings()`](quantization.html#sentence_transformers.quantization.quantize_embeddings)

*

[` semantic_search_faiss() `](quantization.html#sentence_transformers.quantization.semantic_search_faiss)

*

[` semantic_search_usearch() `](quantization.html#sentence_transformers.quantization.semantic_search_usearch)

[Previous](../../examples/training/ms_marco/cross_encoder_README.html "MS MARCO") [Next](SentenceTransformer.html "SentenceTransformer")

* * *

(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).