

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

[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](../distillation/README.html)
- * [Knowledge Distillation](../distillation/README.html#knowledge-distillation)

- * [Speed - Performance Trade-Off](../distillation/README.html#speed-performance-trade-off)
- * [Dimensionality Reduction](../distillation/README.html#dimensionality-reduction)
- * [Quantization](../distillation/README.html#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)]

* Cross-Encoders

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

* * *

Cross-Encoders

SentenceTransformers also supports the option to train Cross-Encoders for sentence pair score and sentence pair classification tasks. For more details on what Cross-Encoders are and the difference between Cross- and Bi-Encoders, see [Cross-Encoders](../applications/cross-encoder/README.html).

Examples

See the following examples how to train Cross-Encoders:

*

[training_stsbenchmark.py](https://github.com/UKPLab/sentence-transformers/tree/master/examples/training/cross-encoder/training_stsbenchmark.py) \- Example how to train for Semantic Textual Similarity (STS) on the STS benchmark dataset.

*

[training_quora_duplicate_questions.py](https://github.com/UKPLab/sentence-transformers/tree/master/examples/training/cross-encoder/training_quora_duplicate_questions.py) \- Example how to train a Cross-Encoder to predict if two questions are duplicates. Uses Quora Duplicate Questions as training dataset.

[training_nli.py](https://github.com/UKPLab/sentence-transformers/tree/master/examples/training/cross-encoder/training_nli.py) \- Example for a multilabel classification task for Natural Language Inference (NLI) task.

Training CrossEncoders

The `CrossEncoder` class is a wrapper around Hugging Face `AutoModelForSequenceClassification`, but with some methods to make training and predicting scores a little bit easier. The saved models are 100% compatible with Hugging Face and can also be loaded with their classes.

First, you need some sentence pair data. You can either have a continuous score, like:

```
from sentence_transformers import InputExample

train_samples = [
    InputExample(texts=["sentence1", "sentence2"], label=0.3),
    InputExample(texts=["Another", "pair"], label=0.8),
]
```

Or you have distinct classes as in the

[training_nli.py](https://github.com/UKPLab/sentence-transformers/tree/master/examples/training/cross-encoder/training_nli.py)
example:

```
from sentence_transformers import InputExample

label2int = {"contradiction": 0, "entailment": 1, "neutral": 2}

train_samples = [
    InputExample(texts=["sentence1", "sentence2"], label=label2int["neutral"]),
    InputExample(texts=["Another", "pair"], label=label2int["entailment"]),
]
```

Then, you define the base model and the number of labels. You can take any [Hugging Face pre-trained model](https://huggingface.co/transformers/pretrained_models.html) that is compatible with AutoModel:

```
model = CrossEncoder('distilroberta-base', num_labels=1)
```

For binary tasks and tasks with continuous scores (like STS), we set num_labels=1. For classification tasks, we set it to the number of labels we

have.

We start the training by calling `model.fit()`:

```
model.fit(  
    train_dataloader=train_dataloader,  
    evaluator=evaluator,  
    epochs=num_epochs,  
    warmup_steps=warmup_steps,  
    output_path=model_save_path,  
)
```

* * *

(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>).