

[ ![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)

\* [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)

\* [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\_mining()'](../../applications/paraphrase-mining/README.html#sentence\_transformers.  
util.paraphrase\_mining)

\* [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](../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 Components

\* Putting It All Together

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

\* [Training Examples](../../docs/sentence\_transformer/training/examples.html)

## \* Hyperparameter Optimization

\* [ Edit on

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

\* \* \*

### # Hyperparameter Optimization

The

[SentenceTransformerTrainer](../docs/package\_reference/sentence\_transformer/trainer.html#sentence\_transformers.trainer.SentenceTransformerTrainer

sentence\_transformers.trainer.SentenceTransformerTrainer

"sentence\_transformers.trainer.SentenceTransformerTrainer") supports

hyperparameter optimization using `transformers`, which in turn supports four

hyperparameter search backends: [optuna](https://optuna.org/),

[sigopt](https://sigopt.org/),

[raytune](https://docs.ray.io/en/latest/tune/index.html), and

[wandb](https://wandb.ai/site/sweeps). You should install your backend of

choice before using it:

```
pip install optuna/sigopt/wandb/ray[tune]
```

On this page, weâ€™ll show you how to use the hyperparameter optimization feature with the optuna backend. The other backends are similar to use, but

you should refer to their respective documentation or the [transformers HPO documentation](https://huggingface.co/docs/transformers/en/hpo\_train) for more information.

## ## HPO Components

The hyperparameter optimization process consists of the following components:

**Hyperparameter Search Space** Specify ranges for hyperparameter values.

**Model Initialization** Initialize a SentenceTransformer model for a trial.

**Loss Initialization** Initialize a loss function given a model.

**Compute Objective** Determines the value to be minimized or maximized.

## ### Hyperparameter Search Space

The hyperparameter search space is defined by a function that returns a dictionary of hyperparameters and their respective search spaces. Here's an example using `optuna` of a search space function that defines the hyperparameters for a SentenceTransformer model:

```
def hpo_search_space(trial):  
    return {  
        "num_train_epochs": trial.suggest_int("num_train_epochs", 1, 2),
```

```

"per_device_train_batch_size": trial.suggest_int("per_device_train_batch_size", 32, 128),
"warmup_ratio": trial.suggest_float("warmup_ratio", 0, 0.3),
"learning_rate": trial.suggest_float("learning_rate", 1e-6, 1e-4, log=True),
}

```

### ### Model Initialization

The model initialization function is a function that takes the hyperparameters of the current `trial` as input and returns a `SentenceTransformer` model. Generally, this function is quite simple. Here's an example of a model initialization function:

```

def hpo_model_init(trial):
    return SentenceTransformer("distilbert-base-uncased")

```

### ### Loss Initialization

The loss initialization function is a function that takes the model initialized for the current trial and returns a loss function. Here's an example of a loss initialization function:

```
def hpo_loss_init(model):

    return losses.CosineSimilarityLoss(model)
```

### ### Compute Objective

The compute objective function is a function that takes the evaluation `metrics` and returns the float value to be minimized or maximized. Here's an example of a compute objective function:

```
def hpo_compute_objective(metrics):

    return metrics["eval_sts-dev_spearman_cosine"]
```

### ## Putting It All Together

You can perform HPO on any regular training loop, the only difference being that you don't call

```
[`SentenceTransformerTrainer.train`](../../docs/package_reference/sentence_transformer/trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.train
"sentence_transformers.trainer.SentenceTransformerTrainer.train"), but
[`SentenceTransformerTrainer.hyperparameter_search`](../../docs/package_reference/sentence_transformer/trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.hyperparameter_search
"sentence_transformers.trainer.SentenceTransformerTrainer.hyperparameter_search")
```

instead. Hereâ€™s an example of how to put it all together:

## Documentation

1. [sentence-transformers/all-nli](https://huggingface.co/datasets/sentence-transformers/all-nli)
2.  
[`EmbeddingSimilarityEvaluator`](../../docs/package\_reference/sentence\_transformer/evaluation.html#sentence\_transformers.evaluation.EmbeddingSimilarityEvaluator  
"sentence\_transformers.evaluation.EmbeddingSimilarityEvaluator")
3. Hyperparameter Search Space
4. Model Initialization
5. Loss Initialization
6. Compute Objective
7.  
[`SentenceTransformerTrainingArguments`](../../docs/package\_reference/sentence\_transformer/training\_args.html#sentence\_transformers.training\_args.SentenceTransformerTrainingArguments  
"sentence\_transformers.training\_args.SentenceTransformerTrainingArguments")
8.  
[`SentenceTransformerTrainer`](../../docs/package\_reference/sentence\_transformer/trainer.html#sentence\_transformers.trainer.SentenceTransformerTrainer



```
"sentence_transformers.trainer.SentenceTransformerTrainer")
```

9.

```
[`hyperparameter_search()](../../docs/package_reference/sentence_transformer/trainer.html#sentence_transformers.trainer.SentenceTransformerTrainer.hyperparameter_search
```

```
"sentence_transformers.trainer.SentenceTransformerTrainer.hyperparameter_search")
```

```
from sentence_transformers import losses

from sentence_transformers import SentenceTransformer, SentenceTransformerTrainer,
SentenceTransformerTrainingArguments

from sentence_transformers.evaluation import EmbeddingSimilarityEvaluator, SimilarityFunction

from sentence_transformers.training_args import BatchSamplers

from datasets import load_dataset

# 1. Load the AllNLI dataset: https://huggingface.co/datasets/sentence-transformers/all-nli, only
10k train and 1k dev

train_dataset = load_dataset("sentence-transformers/all-nli", "triplet", split="train[:10000]")
eval_dataset = load_dataset("sentence-transformers/all-nli", "triplet", split="dev[:1000]")

# 2. Create an evaluator to perform useful HPO

stsb_eval_dataset = load_dataset("sentence-transformers/stsb", split="validation")
dev_evaluator = EmbeddingSimilarityEvaluator(
    sentences1=stsb_eval_dataset["sentence1"],
    sentences2=stsb_eval_dataset["sentence2"],
    scores=stsb_eval_dataset["score"],
```

```
main_similarity=SimilarityFunction.COSINE,  
name="sts-dev",  
)
```

### # 3. Define the Hyperparameter Search Space

```
def hpo_search_space(trial):  
    return {  
        "num_train_epochs": trial.suggest_int("num_train_epochs", 1, 2),  
        "per_device_train_batch_size": trial.suggest_int("per_device_train_batch_size", 32, 128),  
        "warmup_ratio": trial.suggest_float("warmup_ratio", 0, 0.3),  
        "learning_rate": trial.suggest_float("learning_rate", 1e-6, 1e-4, log=True),  
    }
```

### # 4. Define the Model Initialization

```
def hpo_model_init(trial):  
    return SentenceTransformer("distilbert-base-uncased")
```

### # 5. Define the Loss Initialization

```
def hpo_loss_init(model):  
    return losses.MultipleNegativesRankingLoss(model)
```

### # 6. Define the Objective Function

```
def hpo_compute_objective(metrics):  
    """  
    Valid keys are: 'eval_loss', 'eval_sts-dev_pearson_cosine', 'eval_sts-dev_spearman_cosine',  
                    'eval_sts-dev_pearson_manhattan', 'eval_sts-dev_spearman_manhattan',  
                    'eval_sts-dev_pearson_euclidean',
```

```

        'eval_sts-dev_spearman_euclidean',    'eval_sts-dev_pearson_dot',
'eval_sts-dev_spearman_dot',
        'eval_sts-dev_pearson_max',    'eval_sts-dev_spearman_max',    'eval_runtime',
'eval_samples_per_second',
        'eval_steps_per_second', 'epoch'

```

due to the evaluator that we're using.

```

"""

```

```

    return metrics["eval_sts-dev_spearman_cosine"]

```

## # 7. Define the training arguments

```

args = SentenceTransformerTrainingArguments(
    # Required parameter:
    output_dir="checkpoints",

    # Optional training parameters:

    # max_steps=10000, # We might want to limit the number of steps for HPO

    fp16=True, # Set to False if you get an error that your GPU can't run on FP16

    bf16=False, # Set to True if you have a GPU that supports BF16

    batch_sampler=BatchSamplers.NO_DUPLICATES, # MultipleNegativesRankingLoss benefits
from no duplicate samples in a batch

    # Optional tracking/debugging parameters:

    eval_strategy="no", # We don't need to evaluate/save during HPO

    save_strategy="no",

    logging_steps=10,

    run_name="hpo", # Will be used in W&B if `wandb` is installed
)

```

# 8. Create the trainer with model\_init rather than model

```
trainer = SentenceTransformerTrainer(  
    model=None,  
    args=args,  
    train_dataset=train_dataset,  
    eval_dataset=eval_dataset,  
    evaluator=dev_evaluator,  
    model_init=hpo_model_init,  
    loss=hpo_loss_init,  
)
```

# 9. Perform the HPO

```
best_trial = trainer.hyperparameter_search(  
    hp_space=hpo_search_space,  
    compute_objective=hpo_compute_objective,  
    n_trials=20,  
    direction="maximize",  
    backend="optuna",  
)  
print(best_trial)
```

[I 2024-05-17 15:10:47,844] Trial 0 finished with value: 0.7889856589698055 and parameters: {'num\_train\_epochs': 1, 'per\_device\_train\_batch\_size': 123, 'warmup\_ratio': 0.07380948785410107, 'learning\_rate': 2.686331417509812e-06}. Best is trial 0 with value: 0.7889856589698055.

[I 2024-05-17 15:12:13,283] Trial 1 finished with value: 0.7927780672090986 and parameters:

{'num\_train\_epochs': 2, 'per\_device\_train\_batch\_size': 69, 'warmup\_ratio': 0.2927897848007451, 'learning\_rate': 5.885372118095137e-06}. Best is trial 1 with value: 0.7927780672090986.

[I 2024-05-17 15:12:43,896] Trial 2 finished with value: 0.7684829743509601 and parameters: {'num\_train\_epochs': 1, 'per\_device\_train\_batch\_size': 114, 'warmup\_ratio': 0.0739429232666916, 'learning\_rate': 7.344415188959276e-05}. Best is trial 1 with value: 0.7927780672090986.

[I 2024-05-17 15:14:49,730] Trial 3 finished with value: 0.7873032743147989 and parameters: {'num\_train\_epochs': 2, 'per\_device\_train\_batch\_size': 43, 'warmup\_ratio': 0.15184370143796674, 'learning\_rate': 9.703232080395476e-06}. Best is trial 1 with value: 0.7927780672090986.

[I 2024-05-17 15:15:39,597] Trial 4 finished with value: 0.7759251781929949 and parameters: {'num\_train\_epochs': 2, 'per\_device\_train\_batch\_size': 127, 'warmup\_ratio': 0.263946220093495, 'learning\_rate': 1.231454337152625e-06}. Best is trial 1 with value: 0.7927780672090986.

[I 2024-05-17 15:17:02,191] Trial 5 finished with value: 0.7964580509886684 and parameters: {'num\_train\_epochs': 1, 'per\_device\_train\_batch\_size': 34, 'warmup\_ratio': 0.2276865359631089, 'learning\_rate': 7.889007438884571e-06}. Best is trial 5 with value: 0.7964580509886684.

[I 2024-05-17 15:18:55,559] Trial 6 finished with value: 0.7901878917859169 and parameters: {'num\_train\_epochs': 2, 'per\_device\_train\_batch\_size': 48, 'warmup\_ratio': 0.23228838664572948, 'learning\_rate': 2.883013292682523e-06}. Best is trial 5 with value: 0.7964580509886684.

[I 2024-05-17 15:20:27,027] Trial 7 finished with value: 0.7935671067660925 and parameters: {'num\_train\_epochs': 2, 'per\_device\_train\_batch\_size': 62, 'warmup\_ratio': 0.22061123927198237, 'learning\_rate': 2.95413457610349e-06}. Best is trial 5 with value: 0.7964580509886684.

[I 2024-05-17 15:22:23,147] Trial 8 finished with value: 0.7848123114933252 and parameters: {'num\_train\_epochs': 2, 'per\_device\_train\_batch\_size': 45, 'warmup\_ratio': 0.23071701022961139, 'learning\_rate': 9.793681667449783e-06}. Best is trial 5 with value: 0.7964580509886684.

[I 2024-05-17 15:22:52,826] Trial 9 finished with value: 0.7909708416168918 and parameters: {'num\_train\_epochs': 1, 'per\_device\_train\_batch\_size': 121, 'warmup\_ratio': 0.22440506724181647, 'learning\_rate': 4.0744671365843346e-05}. Best is trial 5 with value: 0.7964580509886684.

[I 2024-05-17 15:23:30,395] Trial 10 finished with value: 0.7928991732385567 and parameters:

{'num\_train\_epochs': 1, 'per\_device\_train\_batch\_size': 89, 'warmup\_ratio': 0.14607293301068847, 'learning\_rate': 2.5557492055039498e-05}. Best is trial 5 with value: 0.7964580509886684.

[I 2024-05-17 15:24:18,024] Trial 11 finished with value: 0.7991870087507459 and parameters: {'num\_train\_epochs': 1, 'per\_device\_train\_batch\_size': 66, 'warmup\_ratio': 0.16886154348739527, 'learning\_rate': 3.705926066938032e-06}. Best is trial 11 with value: 0.7991870087507459.

[I 2024-05-17 15:25:44,198] Trial 12 finished with value: 0.7923304174306207 and parameters: {'num\_train\_epochs': 1, 'per\_device\_train\_batch\_size': 33, 'warmup\_ratio': 0.15953772535423974, 'learning\_rate': 1.8076298025704224e-05}. Best is trial 11 with value: 0.7991870087507459.

[I 2024-05-17 15:26:20,739] Trial 13 finished with value: 0.8020260244040395 and parameters: {'num\_train\_epochs': 1, 'per\_device\_train\_batch\_size': 90, 'warmup\_ratio': 0.18105202625281253, 'learning\_rate': 5.513908793512551e-06}. Best is trial 13 with value: 0.8020260244040395.

[I 2024-05-17 15:26:57,783] Trial 14 finished with value: 0.7571110256860063 and parameters: {'num\_train\_epochs': 1, 'per\_device\_train\_batch\_size': 95, 'warmup\_ratio': 0.00122391151793258, 'learning\_rate': 1.0432486633629492e-06}. Best is trial 13 with value: 0.8020260244040395.

[I 2024-05-17 15:27:32,581] Trial 15 finished with value: 0.8009013936824717 and parameters: {'num\_train\_epochs': 1, 'per\_device\_train\_batch\_size': 101, 'warmup\_ratio': 0.1761274711346081, 'learning\_rate': 4.5918293464430035e-06}. Best is trial 13 with value: 0.8020260244040395.

[I 2024-05-17 15:28:05,850] Trial 16 finished with value: 0.8017668050806169 and parameters: {'num\_train\_epochs': 1, 'per\_device\_train\_batch\_size': 103, 'warmup\_ratio': 0.10766501647726355, 'learning\_rate': 5.0309795522333e-06}. Best is trial 13 with value: 0.8020260244040395.

[I 2024-05-17 15:28:37,393] Trial 17 finished with value: 0.7769412380909586 and parameters: {'num\_train\_epochs': 1, 'per\_device\_train\_batch\_size': 108, 'warmup\_ratio': 0.1036610178950246, 'learning\_rate': 1.7747598626081271e-06}. Best is trial 13 with value: 0.8020260244040395.

[I 2024-05-17 15:29:19,340] Trial 18 finished with value: 0.8011921300048339 and parameters: {'num\_train\_epochs': 1, 'per\_device\_train\_batch\_size': 80, 'warmup\_ratio': 0.117014165550441, 'learning\_rate': 1.238558867958792e-05}. Best is trial 13 with value: 0.8020260244040395.

[I 2024-05-17 15:29:59,508] Trial 19 finished with value: 0.8027501854704168 and parameters:

```
{'num_train_epochs': 1, 'per_device_train_batch_size': 84, 'warmup_ratio': 0.014601112207929548,
'learning_rate': 5.627813947769514e-06}. Best is trial 19 with value: 0.8027501854704168.
```

```
BestRun(run_id='19', objective=0.8027501854704168, hyperparameters={'num_train_epochs': 1,
'per_device_train_batch_size': 84, 'warmup_ratio': 0.014601112207929548, 'learning_rate':
5.627813947769514e-06}, run_summary=None)
```

As you can see, the strongest hyperparameters reached **0.802** Spearman correlation on the STS (dev) benchmark. For context, training with the default training arguments (`per_device_train_batch_size=8`, `learning_rate=5e-5`) results in **0.736**, and hyperparameters chosen based on experience (`per_device_train_batch_size=64`, `learning_rate=2e-5`) results in **0.783** Spearman correlation. Consequently, HPO proved quite effective here in improving the model performance.

## ## Example Scripts

- \* `[hpo_nli.py](hpo_nli.py)` \- An example script that performs hyperparameter optimization on the AllNLI dataset.

[ [Previous](#)](../domain\_adaptation/README.html "Domain Adaptation") [ [Next](#)](../docs/sentence\_transformer/training/distributed.html "Distributed Training")

\*\*\*

(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).