

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

## Getting Started

- \* Installation
  - \* Install with pip
  - \* Install with Conda
  - \* Install from Source
  - \* Editable Install
  - \* 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)

- \* [Input Sequence 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-retrieva](#)

[l\)](#)

\* [\[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\(\)\]\(../examples/applications/paraphrase-mining/README.html#sentence\\_trans](#)  
[formers.util.paraphrase\\_mining\)](#)

\* [\[Translated Sentence Mining\]\(../examples/applications/parallel-sentence-mining/README.html\)](#)

\* [\[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#multiplenegativesrankingloss)

\* [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](package\_reference/sentence\_transformer/index.html)
- \* [SentenceTransformer](package\_reference/sentence\_transformer/SentenceTransformer.html)

\*

[SentenceTransformer](package\_reference/sentence\_transformer/SentenceTransformer.html#id1)

\*

[SentenceTransformerModelCardData](package\_reference/sentence\_transformer/SentenceTransformerModelCardData.html#sentencetransformermodelcarddata)

\*

[SimilarityFunction](package\_reference/sentence\_transformer/SentenceTransformer.html#similarityfunction)

- \* [Trainer](package\_reference/sentence\_transformer/trainer.html)

\*

[SentenceTransformerTrainer](package\_reference/sentence\_transformer/trainer.html#sentencetransformertrainer)

\* [Training Arguments](package\_reference/sentence\_transformer/training\_args.html)

\*

[SentenceTransformerTrainingArguments](package\_reference/sentence\_transformer/training\_args.html#sentencetransformertrainingarguments)

\* [Losses](package\_reference/sentence\_transformer/losses.html)

\* [BatchAllTripletLoss](package\_reference/sentence\_transformer/losses.html#batchalltripletloss)

\*

[BatchHardSoftMarginTripletLoss](package\_reference/sentence\_transformer/losses.html#batchhardsoftmargintripletloss)

\*

[BatchHardTripletLoss](package\_reference/sentence\_transformer/losses.html#batchhardtripletloss)

\*

[BatchSemiHardTripletLoss](package\_reference/sentence\_transformer/losses.html#batchsemihardtripletloss)

\* [ContrastiveLoss](package\_reference/sentence\_transformer/losses.html#contrastiveloss)

\*

[OnlineContrastiveLoss](package\_reference/sentence\_transformer/losses.html#onlinecontrastiveloss)

\*

[ContrastiveTensionLoss](package\_reference/sentence\_transformer/losses.html#contrastivetensionloss)

\*

[ContrastiveTensionLossInBatchNegatives](package\_reference/sentence\_transformer/losses.html#contrastivetensionlossinbatchnegatives)

\* [CoSENTLoss](package\_reference/sentence\_transformer/losses.html#cosentloss)

\* [AngleLoss](package\_reference/sentence\_transformer/losses.html#angleloss)

\*

[CosineSimilarityLoss](package\_reference/sentence\_transformer/losses.html#cosinesimilarityloss)

\*

[DenoisingAutoEncoderLoss](package\_reference/sentence\_transformer/losses.html#denoisingautoencoderloss)

\* [GISTEmbedLoss](package\_reference/sentence\_transformer/losses.html#gistembedloss)

\*

[CachedGISTEmbedLoss](package\_reference/sentence\_transformer/losses.html#cachedgistembedloss)

\* [MSELoss](package\_reference/sentence\_transformer/losses.html#mseloss)

\* [MarginMSELoss](package\_reference/sentence\_transformer/losses.html#marginmseloss)

\* [MatryoshkaLoss](package\_reference/sentence\_transformer/losses.html#matryoshkaloss)

\* [Matryoshka2dLoss](package\_reference/sentence\_transformer/losses.html#matryoshka2dloss)

\* [AdaptiveLayerLoss](package\_reference/sentence\_transformer/losses.html#adaptivelayerloss)

\*

[MegaBatchMarginLoss](package\_reference/sentence\_transformer/losses.html#megabatchmarginloss)

\*

[MultipleNegativesRankingLoss](package\_reference/sentence\_transformer/losses.html#multiplenegativesrankingloss)

\*

[CachedMultipleNegativesRankingLoss](package\_reference/sentence\_transformer/losses.html#cachedmultiplenegativesrankingloss)

\*

[MultipleNegativesSymmetricRankingLoss](package\_reference/sentence\_transformer/losses.html#multiplenegativessymmetricrankingloss)

\*

[CachedMultipleNegativesSymmetricRankingLoss](package\_reference/sentence\_transformer/losses.html#cachedmultiplenegativessymmetricrankingloss)

\* [SoftmaxLoss](package\_reference/sentence\_transformer/losses.html#softmaxloss)

\* [TripletLoss](package\_reference/sentence\_transformer/losses.html#tripletloss)

\* [Samplers](package\_reference/sentence\_transformer/sampler.html)

\* [BatchSamplers](package\_reference/sentence\_transformer/sampler.html#batchsamplers)

\*

[MultiDatasetBatchSamplers](package\_reference/sentence\_transformer/sampler.html#multidatasetbatchsamplers)

\* [Evaluation](package\_reference/sentence\_transformer/evaluation.html)

\*

[BinaryClassificationEvaluator](package\_reference/sentence\_transformer/evaluation.html#binaryclassificationevaluator)

\*

[EmbeddingSimilarityEvaluator](package\_reference/sentence\_transformer/evaluation.html#embeddingsimilarityevaluator)

\*

[InformationRetrievalEvaluator](package\_reference/sentence\_transformer/evaluation.html#informationretrievalevaluator)

\*

[NanoBEIREvaluator](package\_reference/sentence\_transformer/evaluation.html#nanobeirevaluator)

\* [MSEEvaluator](package\_reference/sentence\_transformer/evaluation.html#mseevaluator)

\*

[ParaphraseMiningEvaluator](package\_reference/sentence\_transformer/evaluation.html#paraphraseminingevaluator)

\*

[RerankingEvaluator](package\_reference/sentence\_transformer/evaluation.html#rerankingevaluator)

\*

[SentenceEvaluator](package\_reference/sentence\_transformer/evaluation.html#sentenceevaluator)

\*

[SequentialEvaluator](package\_reference/sentence\_transformer/evaluation.html#sequentialevaluator)

\*

[TranslationEvaluator](package\_reference/sentence\_transformer/evaluation.html#translationevaluator)

\* [TripletEvaluator](package\_reference/sentence\_transformer/evaluation.html#tripletevaluator)

\* [Datasets](package\_reference/sentence\_transformer/datasets.html)

\*

[ParallelSentencesDataset](package\_reference/sentence\_transformer/datasets.html#parallelsentencesdataset)

\*

[SentenceLabelDataset](package\_reference/sentence\_transformer/datasets.html#sentencelabeldataset)

\*

[DenoisingAutoEncoderDataset](package\_reference/sentence\_transformer/datasets.html#denoisingautoencoderdataset)

\*

[NoDuplicatesDataLoader](package\_reference/sentence\_transformer/datasets.html#noduplicatesdataloader)

\* [Models](package\_reference/sentence\_transformer/models.html)

\* [Main Classes](package\_reference/sentence\_transformer/models.html#main-classes)

\* [Further Classes](package\_reference/sentence\_transformer/models.html#further-classes)

\* [quantization](package\_reference/sentence\_transformer/quantization.html)

\*

[`quantize\_embeddings()`](package\_reference/sentence\_transformer/quantization.html#sentence\_transformers.quantization.quantize\_embeddings)

\*

[`semantic\_search\_faiss()`](package\_reference/sentence\_transformer/quantization.html#sentence\_transformers.quantization.semantic\_search\_faiss)

\*

[`semantic\_search\_usearch()`](package\_reference/sentence\_transformer/quantization.html#sentence\_transformers.quantization.semantic\_search\_usearch)

\* [Cross Encoder](package\_reference/cross\_encoder/index.html)

\* [CrossEncoder](package\_reference/cross\_encoder/cross\_encoder.html)

\* [CrossEncoder](package\_reference/cross\_encoder/cross\_encoder.html#id1)

\* [Training Inputs](package\_reference/cross\_encoder/cross\_encoder.html#training-inputs)

\* [Evaluation](package\_reference/cross\_encoder/evaluation.html)

\*

[CEBinaryAccuracyEvaluator](package\_reference/cross\_encoder/evaluation.html#cebinaryaccuracyevaluator)

\*

[CEBinaryClassificationEvaluator](package\_reference/cross\_encoder/evaluation.html#cebinaryclassificationevaluator)

\*

[CECorrelationEvaluator](package\_reference/cross\_encoder/evaluation.html#cecorrelationevaluator)

\* [CEF1Evaluator](package\_reference/cross\_encoder/evaluation.html#cef1evaluator)

\*

[CESoftmaxAccuracyEvaluator](package\_reference/cross\_encoder/evaluation.html#cesoftmaxaccuracyevaluator)



\*

[CERerankingEvaluator](package\_reference/cross\_encoder/evaluation.html#cererankingevaluator)

\* [util](package\_reference/util.html)

\* [Helper Functions](package\_reference/util.html#module-sentence\_transformers.util)

\*

[`community\_detection()`](package\_reference/util.html#sentence\_transformers.util.community\_detection)

\* [`http\_get()`](package\_reference/util.html#sentence\_transformers.util.http\_get)

\*

[`is\_training\_available()`](package\_reference/util.html#sentence\_transformers.util.is\_training\_available)

\*

[`mine\_hard\_negatives()`](package\_reference/util.html#sentence\_transformers.util.mine\_hard\_negatives)

\*

[`normalize\_embeddings()`](package\_reference/util.html#sentence\_transformers.util.normalize\_embeddings)

\*

[`paraphrase\_mining()`](package\_reference/util.html#sentence\_transformers.util.paraphrase\_mining)

\*

[`semantic\_search()`](package\_reference/util.html#sentence\_transformers.util.semantic\_search)

\*

[`truncate\_embeddings()`](package\_reference/util.html#sentence\_transformers.util.truncate\_embeddings)

\* [Model Optimization](package\_reference/util.html#module-sentence\_transformers.backend)

\*

[`export\_dynamic\_quantized\_onnx\_model()`](package\_reference/util.html#sentence\_transformers.backend.export\_dynamic\_quantized\_onnx\_model)

\*

[`export\_optimized\_onnx\_model()`](package\_reference/util.html#sentence\_transformers.backend.export\_optimized\_onnx\_model)

\*

[`export\_static\_quantized\_openvino\_model()`](package\_reference/util.html#sentence\_transformers.backend.export\_static\_quantized\_openvino\_model)

\* [Similarity Metrics](package\_reference/util.html#module-sentence\_transformers.util)

\* [`cos\_sim()`](package\_reference/util.html#sentence\_transformers.util.cos\_sim)

\* [`dot\_score()`](package\_reference/util.html#sentence\_transformers.util.dot\_score)

\* [`euclidean\_sim()`](package\_reference/util.html#sentence\_transformers.util.euclidean\_sim)

\* [`manhattan\_sim()`](package\_reference/util.html#sentence\_transformers.util.manhattan\_sim)

\*

[`pairwise\_cos\_sim()`](package\_reference/util.html#sentence\_transformers.util.pairwise\_cos\_sim)

\*

[`pairwise\_dot\_score()`](package\_reference/util.html#sentence\_transformers.util.pairwise\_dot\_score)

\*

[`pairwise\_euclidean\_sim()`](package\_reference/util.html#sentence\_transformers.util.pairwise\_euclidean\_sim)

\*

[`pairwise\_manhattan\_sim()`](package\_reference/util.html#sentence\_transformers.util.pairwise\_manhattan\_sim)

\_\_[Sentence Transformers](../index.html)

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

\* Installation

\* [ Edit on

GitHub](https://github.com/UKPLab/sentence-transformers/blob/master/docs/installation.md)

\* \* \*

# Installation

We recommend **Python 3.9+** , **[PyTorch 1.11.0+](https://pytorch.org/get-started/locally/)** , and **[transformers v4.11.0+](https://github.com/huggingface/transformers)** . There are 5 extra options to install Sentence Transformers:

\* **Default:** This allows for loading, saving, and inference (i.e., getting embeddings) of models.

\* **ONNX:** This allows for loading, saving, inference, optimizing, and quantizing of models using the ONNX backend.

\* **OpenVINO:** This allows for loading, saving, and inference of models using the OpenVINO backend.

\* **Default and Training** : Like **Default** , plus training.

\* **Development** : All of the above plus some dependencies for developing Sentence Transformers, see Editable Install.

Note that you can mix and match the various extras, e.g. ``pip install -U`

`"sentence-transformers[train,onnx-gpu]"``.

## Install with pip•

Default

```
pip install -U sentence-transformers
```

ONNX

For GPU and CPU:

```
pip install -U "sentence-transformers[onnx-gpu]"
```

For CPU only:

```
pip install -U "sentence-transformers[onnx]"
```

## OpenVINO

```
pip install -U "sentence-transformers[openvino]"
```

## Default and Training

```
pip install -U "sentence-transformers[train]"
```

To use [Weights and Biases](<https://wandb.ai/>) to track your training logs, you should also install `wandb` **(recommended)** :

```
pip install wandb
```

And to track your Carbon Emissions while training and have this information automatically included in your model cards, also install `codecarbon`

**(recommended)** :

```
pip install codecarbon
```

## Development

```
pip install -U "sentence-transformers[dev]"
```

## Install with Conda

## Default

```
conda install -c conda-forge sentence-transformers
```

## ONNX

For GPU and CPU:

```
pip install -U "sentence-transformers[onnx-gpu]"
```

For CPU only:

```
pip install -U "sentence-transformers[onnx]"
```

OpenVINO

```
pip install -U "sentence-transformers[openvino]"
```

Default and Training

```
conda install -c conda-forge sentence-transformers accelerate datasets
```

To use [Weights and Biases](<https://wandb.ai/>) to track your training logs,

you should also install `wandb` **(recommended)** :

```
pip install wandb
```

And to track your Carbon Emissions while training and have this information automatically included in your model cards, also install `codecarbon`

**(recommended)** :

```
pip install codecarbon
```

## Development

```
conda install -c conda-forge sentence-transformers accelerate datasets pre-commit pytest ruff
```

### ## Install from Source

You can install `sentence-transformers` directly from source to take advantage of the bleeding edge `master` branch rather than the latest stable release:

## Default



```
pip install git+https://github.com/UKPLab/sentence-transformers.git
```

ONNX

For GPU and CPU:

```
pip install -U "sentence-transformers[onnx-gpu]" @  
git+https://github.com/UKPLab/sentence-transformers.git"
```

For CPU only:

```
pip install -U "sentence-transformers[onnx]" @  
git+https://github.com/UKPLab/sentence-transformers.git"
```

OpenVINO

```
pip install -U "sentence-transformers[openvino] @  
git+https://github.com/UKPLab/sentence-transformers.git"
```

## Default and Training

```
pip install -U "sentence-transformers[train] @  
git+https://github.com/UKPLab/sentence-transformers.git"
```

To use [Weights and Biases](<https://wandb.ai/>) to track your training logs,  
you should also install `wandb` **(recommended)** :

```
pip install wandb
```

And to track your carbon emissions while training and have this information  
automatically included in your model cards, also install `codecarbon`  
**(recommended)** :

```
pip install codecarbon
```

## Development

```
pip install -U "sentence-transformers[dev] @  
git+https://github.com/UKPLab/sentence-transformers.git"
```

### ## Editable Install

If you want to make changes to `sentence-transformers`, you will need an editable install. Clone the repository and install it with these commands:

```
git clone https://github.com/UKPLab/sentence-transformers  
cd sentence-transformers  
pip install -e ".[train,dev]"
```

These commands will link the new `sentence-transformers` folder and your Python library paths, such that this folder will be used when importing `sentence-transformers`.

## ## Install PyTorch with CUDA support

To use a GPU/CUDA, you must install PyTorch with CUDA support. Follow [PyTorch - Get Started](https://pytorch.org/get-started/locally/) for installation steps.

[ Previous](../index.html "SentenceTransformers Documentation") [Next](quickstart.html "Quickstart")

\* \* \*

(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).