

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

## 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

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

\* [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](docs/sentence\_transformer/usage/semantic\_textual\_similarity.html)

\* [Similarity Calculation](docs/sentence\_transformer/usage/semantic\_textual\_similarity.html#similarity-calculation)

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

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

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

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

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

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

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

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

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

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

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

\* [Retrieve & Re-Rank]

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

\* [Retrieval:

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

\* [Re-Ranker:

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

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

\* [Pre-trained Bi-Encoders

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

\* [Pre-trained Cross-Encoders

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

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

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

\* [Agglomerative

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

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

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

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

\*

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

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

\* [Margin Based

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

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

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

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

- \* [Usage](examples/applications/image-search/README.html#usage)
- \* [Examples](examples/applications/image-search/README.html#examples)
- \* [Embedding Quantization](examples/applications/embedding-quantization/README.html)
  - \* [Binary Quantization](examples/applications/embedding-quantization/README.html#binary-quantization)
    - \* [Scalar (int8) Quantization](examples/applications/embedding-quantization/README.html#scalar-int8-quantization)
    - \* [Additional extensions](examples/applications/embedding-quantization/README.html#additional-extensions)
  - \* [Demo](examples/applications/embedding-quantization/README.html#demo)
  - \* [Try it yourself](examples/applications/embedding-quantization/README.html#try-it-yourself)
- \* [Speeding up Inference](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](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-learnin

g)

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

\* [GPL: Generative

Pseudo-Labeling](examples/domain\_adaptation/README.html#gpu-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](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](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](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](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](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#simil

arityfunction)

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

\*

[SentenceTransformerTrainer](docs/package\_reference/sentence\_transformer/trainer.html#sentence-transformer-trainer)

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

\*

[SentenceTransformerTrainingArguments](docs/package\_reference/sentence\_transformer/training\_args.html#sentence-transformer-training-arguments)

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

\*

[BatchAllTripletLoss](docs/package\_reference/sentence\_transformer/losses.html#batch-all-triplet-loss)

\*

[BatchHardSoftMarginTripletLoss](docs/package\_reference/sentence\_transformer/losses.html#batch-hard-soft-margin-triplet-loss)

\*

[BatchHardTripletLoss](docs/package\_reference/sentence\_transformer/losses.html#batch-hard-triplet-loss)

\*

[BatchSemiHardTripletLoss](docs/package\_reference/sentence\_transformer/losses.html#batch-semi-hard-triplet-loss)

\* [ContrastiveLoss](docs/package\_reference/sentence\_transformer/losses.html#contrastive-loss)

\*

[OnlineContrastiveLoss](docs/package\_reference/sentence\_transformer/losses.html#online-contrastive-loss)

\*

[ContrastiveTensionLoss](docs/package\_reference/sentence\_transformer/losses.html#contrastive-tension-loss)

nsionloss)

\*

[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)

\* [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)

rginloss)

\*

[MultipleNegativesRankingLoss](docs/package\_reference/sentence\_transformer/losses.html#multiple-negatives-ranking-loss)

\*

[CachedMultipleNegativesRankingLoss](docs/package\_reference/sentence\_transformer/losses.html#cached-multiple-negatives-ranking-loss)

\*

[MultipleNegativesSymmetricRankingLoss](docs/package\_reference/sentence\_transformer/losses.html#multiple-negatives-symmetric-ranking-loss)

\*

[CachedMultipleNegativesSymmetricRankingLoss](docs/package\_reference/sentence\_transformer/losses.html#cached-multiple-negatives-symmetric-ranking-loss)

\* [SoftmaxLoss](docs/package\_reference/sentence\_transformer/losses.html#softmax-loss)

\* [TripletLoss](docs/package\_reference/sentence\_transformer/losses.html#triplet-loss)

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

\* [BatchSamplers](docs/package\_reference/sentence\_transformer/sampler.html#batch-samplers)

\*

[MultiDatasetBatchSamplers](docs/package\_reference/sentence\_transformer/sampler.html#multi-dataset-batch-samplers)

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

\*

[BinaryClassificationEvaluator](docs/package\_reference/sentence\_transformer/evaluation.html#binary-classification-evaluator)

\*

[EmbeddingSimilarityEvaluator](docs/package\_reference/sentence\_transformer/evaluation.html#embedding-similarity-evaluator)

\*

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

\*

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

\*

[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#rerankingevaluator)

\*

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

\*

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

\*

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

\*

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

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

\*

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

\*

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

\*

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

\*

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

- \* [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\_transformers.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

\* \* SentenceTransformers Documentation

\* [ [Edit on GitHub](https://github.com/UKPLab/sentence-transformers/blob/master/index.rst)](https://github.com/UKPLab/sentence-transformers/blob/master/index.rst)

\* \* \*

Tip

You can now use ONNX and OpenVINO backends for Sentence Transformer models.

Read [SentenceTransformer > Usage > Speeding up


Inference](docs/sentence\_transformer/usage/efficiency.html) to learn more about the new backends and what they can mean for your inference speed.


### Tip

Sentence Transformers v3.4 just released, introducing documentation for training with PEFT. Read [SentenceTransformer > Training Examples > Training with PEFT Adapters](examples/training/peft/README.html) to learn more about how you can use train embedding models without finetuning all model parameters.

### # SentenceTransformers Documentation

Sentence Transformers (a.k.a. SBERT) is the go-to Python module for accessing, using, and training state-of-the-art text and image embedding models. It can be used to compute embeddings using Sentence Transformer models ([quickstart](docs/quickstart.html#sentence-transformer)) or to calculate similarity scores using Cross-Encoder models ([quickstart](docs/quickstart.html#cross-encoder)). This unlocks a wide range of applications, including [semantic search](examples/applications/semantic-search/README.html), [semantic textual similarity](docs/usage/semantic\_textual\_similarity.html), and [paraphrase mining](examples/applications/paraphrase-mining/README.html).

A wide selection of over [5,000 pre-trained Sentence Transformers models](https://huggingface.co/models?library=sentence-transformers) are available for immediate use on  Hugging Face, including many of the state-of-the-art models from the [Massive Text Embeddings Benchmark (MTEB) leaderboard](https://huggingface.co/spaces/mteb/leaderboard). Additionally, it is easy to [train or finetune your own models](docs/sentence\_transformer/training\_overview.html) using Sentence Transformers, enabling you to create custom models for your specific use cases.

Sentence Transformers was created by [UKPLab](http://www.ukp.tu-darmstadt.de/) and is being maintained by  Hugging Face](https://huggingface.co). Donâ€™t hesitate to open an issue on the [Sentence Transformers repository](https://github.com/UKPLab/sentence-transformers) if something is broken or if you have further questions.

## # Usage

See also

See the [Quickstart](docs/quickstart.html) for more quick information on how to use Sentence Transformers.

Using Sentence Transformer models is elementary:

Installation

You can install `_sentence-transformers_` using pip:

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

We recommend **Python 3.9+** and **PyTorch 1.11.0+**. See [\[installation\]\(docs/installation.html\)](#) for further installation options.

```
from sentence_transformers import SentenceTransformer
```

```
# 1. Load a pretrained Sentence Transformer model
```

```
model = SentenceTransformer("all-MiniLM-L6-v2")
```

```
# The sentences to encode
```

```
sentences = [
```

```
    "The weather is lovely today.",
```

```
    "It's so sunny outside!",
```

```
    "He drove to the stadium.",
```

```
]
```

```
# 2. Calculate embeddings by calling model.encode()
```

```
embeddings = model.encode(sentences)
```

```
print(embeddings.shape)
```

# [3, 384]

# 3. Calculate the embedding similarities

```
similarities = model.similarity(embeddings, embeddings)
```

```
print(similarities)
```

```
# tensor([[1.0000, 0.6660, 0.1046],
```

```
#      [0.6660, 1.0000, 0.1411],
```

```
#      [0.1046, 0.1411, 1.0000]])
```

# What Next? *if*•

Consider reading one of the following sections to answer the related questions:

- \* How to **use** Sentence Transformer models? [Sentence Transformers > Usage](docs/sentence\_transformer/usage/usage.html)

- \* What Sentence Transformer **models** can I use? [Sentence Transformers > Pretrained Models](docs/sentence\_transformer/pretrained\_models.html)

- \* How do I make Sentence Transformer models **faster**? [Sentence Transformers > Usage > Speeding up Inference](docs/sentence\_transformer/usage/efficiency.html)

- \* How do I **train/finetune** a Sentence Transformer model? [Sentence Transformers > Training Overview](docs/sentence\_transformer/training\_overview.html)

\* How to **use** Cross Encoder models? [Cross Encoder > Usage](docs/cross\_encoder/usage/usage.html)

\* What Cross Encoder **models** can I use? [Cross Encoder > Pretrained Models](docs/cross\_encoder/pretrained\_models.html)

# Citing

If you find this repository helpful, feel free to cite our publication

[Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks](https://arxiv.org/abs/1908.10084):

```
>
> @inproceedings{reimers-2019-sentence-bert,
>   title = "Sentence-BERT: Sentence Embeddings using Siamese BERT-
> Networks",
>   author = "Reimers, Nils and Gurevych, Iryna",
>   booktitle = "Proceedings of the 2019 Conference on Empirical Methods
> in Natural Language Processing",
>   month = "11",
>   year = "2019",
>   publisher = "Association for Computational Linguistics",
>   url = "https://arxiv.org/abs/1908.10084",
> }
>
```

If you use one of the multilingual models, feel free to cite our publication



[Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation](<https://arxiv.org/abs/2004.09813>):

```
>
> @inproceedings{reimers-2020-multilingual-sentence-bert,
>   title = "Making Monolingual Sentence Embeddings Multilingual using
> Knowledge Distillation",
>   author = "Reimers, Nils and Gurevych, Iryna",
>   booktitle = "Proceedings of the 2020 Conference on Empirical Methods
> in Natural Language Processing",
>   month = "11",
>   year = "2020",
>   publisher = "Association for Computational Linguistics",
>   url = "https://arxiv.org/abs/2004.09813",
> }
>
```

If you use the code for [data

augmentation]([https://github.com/UKPLab/sentence-](https://github.com/UKPLab/sentence-transformers/tree/master/examples/training/data_augmentation)

[transformers/tree/master/examples/training/data\\_augmentation](https://github.com/UKPLab/sentence-transformers/tree/master/examples/training/data_augmentation)), feel free to

cite our publication [Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring

Tasks](<https://arxiv.org/abs/2010.08240>):

```
>
> @inproceedings{thakur-2020-AugSBERT,
>   title = "Augmented {SBERT}: Data Augmentation Method for Improving Bi-
```

```
> Encoders for Pairwise Sentence Scoring Tasks",
>     author = "Thakur, Nandan and Reimers, Nils and Daxenberger, Johannes
> and Gurevych, Iryna",
>     booktitle = "Proceedings of the 2021 Conference of the North American
> Chapter of the Association for Computational Linguistics: Human Language
> Technologies",
>     month = jun,
>     year = "2021",
>     address = "Online",
>     publisher = "Association for Computational Linguistics",
>     url = "https://www.aclweb.org/anthology/2021.naacl-main.28",
>     pages = "296--310",
> }
>
```

[Next ]([docs/installation.html](#) "Installation")

\* \* \*

(C) Copyright 2025.

Built with [Sphinx](<https://www.sphinx-doc.org/>) using a

[theme]([https://github.com/readthedocs/sphinx\\_rtd\\_theme](https://github.com/readthedocs/sphinx_rtd_theme)) provided by [Read the Docs](<https://readthedocs.org>).