

![[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#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](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) provided by [Read the Docs](<https://readthedocs.org>).