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

Getting Started

* [Installation](../installation.html)
  * [Install with pip](../installation.html#install-with-pip)
  * [Install with Conda](../installation.html#install-with-conda)
  * [Install from Source](../installation.html#install-from-source)
  * [Editable Install](../installation.html#editable-install)
  * [Install PyTorch with CUDA support](../installation.html#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](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](usage/semantic_textual_similarity.html)
* [Similarity Calculation](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](usage/efficiency.html)
    * [PyTorch](usage/efficiency.html#pytorch)
    * [ONNX](usage/efficiency.html#onnx)
    * [OpenVINO](usage/efficiency.html#openvino)
    * [Benchmarks](usage/efficiency.html#benchmarks)
  * [Creating Custom Models](usage/custom_models.html)
    * [Structure of Sentence Transformer Models](usage/custom_models.html#structure-of-sentence-transformer-models)

* [Sentence Transformer Model from a Transformers Model](usage/custom_models.html#sentence-transformer-model-from-a-transformers-model)
* [Pretrained Models](pretrained_models.html)
  * [Original Models](pretrained_models.html#original-models)
  * [Semantic Search Models](pretrained_models.html#semantic-search-models)
    * [Multi-QA Models](pretrained_models.html#multi-qa-models)
    * [MSMARCO Passage Models](pretrained_models.html#msmarco-passage-models)
  * [Multilingual Models](pretrained_models.html#multilingual-models)
    * [Semantic Similarity Models](pretrained_models.html#semantic-similarity-models)
    * [Bitext Mining](pretrained_models.html#bitext-mining)
  * [Image & Text-Models](pretrained_models.html#image-text-models)
  * [INSTRUCTOR models](pretrained_models.html#instructor-models)
  * [Scientific Similarity Models](pretrained_models.html#scientific-similarity-models)
* [Training Overview](training_overview.html)
  * [Why Finetune?](training_overview.html#why-finetune)
  * [Training Components](training_overview.html#training-components)
  * [Dataset](training_overview.html#dataset)
    * [Dataset Format](training_overview.html#dataset-format)
  * [Loss Function](training_overview.html#loss-function)
  * [Training Arguments](training_overview.html#training-arguments)
  * [Evaluator](training_overview.html#evaluator)
  * [Trainer](training_overview.html#trainer)
    * [Callbacks](training_overview.html#callbacks)
  * [Multi-Dataset Training](training_overview.html#multi-dataset-training)
  * [Deprecated Training](training_overview.html#deprecated-training)
  * [Best Base Embedding Models](training_overview.html#best-base-embedding-models)
* Dataset Overview

    * Datasets on the Hugging Face Hub
    * Pre-existing Datasets
  * [Loss Overview](loss_overview.html)
    * [Loss modifiers](loss_overview.html#loss-modifiers)
    * [Distillation](loss_overview.html#distillation)
    * [Commonly used Loss Functions](loss_overview.html#commonly-used-loss-functions)
    * [Custom Loss Functions](loss_overview.html#custom-loss-functions)
  * [Training Examples](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](training/distributed.html)
  * [Comparison](training/distributed.html#comparison)
  * [FSDP](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/SentenceTransformer.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#noduplicatesd

ataloader)
    * [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#cecorrelationevaluat

or)
    * [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)
  * Dataset Overview
  * [Edit on GitHub](https://github.com/UKPLab/sentence-transformers/blob/master/docs/sentence_transformer/dataset_overview.md)

* * *

Dataset Overviewïƒ•

Hint

Quickstart: Find [curated datasets](https://huggingface.co/collections/sentence-transformers/embedding-model-datasets-6644d7a3673a511914aa7552) or [community datasets](https://huggingface.co/datasets?other=sentence-transformers), choose a loss function via this [loss overview](loss_overview.html), and [verify](training_overview.html#dataset-format) that it works with your dataset.

It is important that your dataset format matches your loss function (or that

you choose a loss function that matches your dataset format). See [Training Overview > Dataset Format](training_overview.html#dataset-format) to learn how to verify whether a dataset format works with a loss function.

In practice, most dataset configurations will take one of four forms:

* **Positive Pair** : A pair of related sentences. This can be used both for symmetric tasks (semantic textual similarity) or asymmetric tasks (semantic search), with examples including pairs of paraphrases, pairs of full texts and their summaries, pairs of duplicate questions, pairs of (`query`, `response`), or pairs of (`source_language`, `target_language`). Natural Language Inference datasets can also be formatted this way by pairing entailing sentences.

  * **Examples:** [sentence-transformers/sentence-compression](https://huggingface.co/datasets/sentence-transformers/sentence-compression), [sentence-transformers/coco-captions](https://huggingface.co/datasets/sentence-transformers/coco-captions), [sentence-transformers/codesearchnet](https://huggingface.co/datasets/sentence-transformers/codesearchnet), [sentence-transformers/natural-questions](https://huggingface.co/datasets/sentence-transformers/natural-questions), [sentence-transformers/gooaq](https://huggingface.co/datasets/sentence-transformers/gooaq), [sentence-transformers/squad](https://huggingface.co/datasets/sentence-transformers/squad), [sentence-transformers/wikihow](https://huggingface.co/datasets/sentence-transformers/wikihow), [sentence-transformers/eli5](https://huggingface.co/datasets/sentence-transformers/eli5)

* **Triplets** : (anchor, positive, negative) text triplets. These datasets donâ€™t need labels.

* **Examples:** [sentence-transformers/quora-duplicates](https://huggingface.co/datasets/sentence-transformers/quora-duplicates), [nirantk/triplets](https://huggingface.co/datasets/nirantk/triplets), [sentence-transformers/all-nli](https://huggingface.co/datasets/sentence-transformers/all-nli)

* **Pair with Similarity Score** : A pair of sentences with a score indicating their similarity. Common examples are â€œSemantic Textual Similarityâ€• datasets.

* **Examples:** [sentence-transformers/stsb](https://huggingface.co/datasets/sentence-transformers/stsb), [PhilipMay/stsb_multi_mt](https://huggingface.co/datasets/PhilipMay/stsb_multi_mt).

* **Texts with Classes** : A text with its corresponding class. This data format is easily converted by loss functions into three sentences (triplets) where the first is an â€œanchorâ€•, the second a â€œpositiveâ€• of the same class as the anchor, and the third a â€œnegativeâ€• of a different class.

* **Examples:** [trec](https://huggingface.co/datasets/trec), [yahoo_answers_topics](https://huggingface.co/datasets/yahoo_answers_topics).

Note that it is often simple to transform a dataset from one format to another, such that it works with your loss function of choice.

Tip

You can use

[`mine_hard_negatives()`](../package_reference/util.html#sentence_transformers.util.mine_hard_negatives
"sentence_transformers.util.mine_hard_negatives") to convert a dataset of
positive pairs into a dataset of triplets. It uses a
[`SentenceTransformer`](../package_reference/sentence_transformer/SentenceTransformer.html#sentence_transformers.SentenceTransformer
"sentence_transformers.SentenceTransformer") model to find hard negatives:
texts that are similar to the first dataset column, but are not quite as
similar as the text in the second dataset column. Datasets with hard triplets
often outperform datasets with just positive pairs.

For example, we mined hard negatives from [sentence-
transformers/gooaq](https://huggingface.co/datasets/sentence-
transformers/gooaq) to produce [tomaarsen/gooaq-hard-
negatives](https://huggingface.co/datasets/tomaarsen/gooaq-hard-negatives) and
trained [tomaarsen/mpnet-base-gooaq](https://huggingface.co/tomaarsen/mpnet-
base-gooaq) and [tomaarsen/mpnet-base-gooaq-hard-
negatives](https://huggingface.co/tomaarsen/mpnet-base-gooaq-hard-negatives)
on the two datasets, respectively. Sadly, the two models use a different
evaluation split, so their performance canâ€™t be compared directly.

Datasets on the Hugging Face Hubïƒ•

The [Datasets library](https://huggingface.co/docs/datasets/index) (`pip
install datasets`) allows you to load datasets from the Hugging Face Hub with
the
[`load_dataset()`](https://huggingface.co/docs/datasets/main/en/package_reference/loading_method

s#datasets.load_dataset

"\(in datasets vmain\)") function:

```
from datasets import load_dataset

# Indicate the dataset id from the Hub
dataset_id = "sentence-transformers/natural-questions"
dataset = load_dataset(dataset_id, split="train")
"""
Dataset({
    features: ['query', 'answer'],
    num_rows: 100231
})
"""
print(dataset[0])
"""
{
    'query': 'when did richmond last play in a preliminary final',
     'answer': "Richmond Football Club Richmond began 2017 with 5 straight wins, a feat it had not achieved since 1995. A series of close losses hampered the Tigers throughout the middle of the season, including a 5-point loss to the Western Bulldogs, 2-point loss to Fremantle, and a 3-point loss to the Giants. Richmond ended the season strongly with convincing victories over Fremantle and St Kilda in the final two rounds, elevating the club to 3rd on the ladder. Richmond's first final of the season against the Cats at the MCG attracted a record qualifying final crowd of 95,028; the Tigers won by 51 points. Having advanced to the first preliminary finals for the first time since 2001,
```

Richmond defeated Greater Western Sydney by 36 points in front of a crowd of 94,258 to progress to the Grand Final against Adelaide, their first Grand Final appearance since 1982. The attendance was 100,021, the largest crowd to a grand final since 1986. The Crows led at quarter time and led by as many as 13, but the Tigers took over the game as it progressed and scored seven straight goals at one point. They eventually would win by 48 points â€“ 16.12 (108) to Adelaide's 8.12 (60) â€“ to end their 37-year flag drought.[22] Dustin Martin also became the first player to win a Premiership medal, the Brownlow Medal and the Norm Smith Medal in the same season, while Damien Hardwick was named AFL Coaches Association Coach of the Year. Richmond's jump from 13th to premiers also marked the biggest jump from one AFL season to the next."

}

"""

For more information on how to manipulate your dataset see the [Datasets Documentation](https://huggingface.co/docs/datasets/access).

Tip

Itâ€™s common for Hugging Face Datasets to contain extraneous columns, e.g. sample_id, metadata, source, type, etc. You can use [`Dataset.remove_columns`](https://huggingface.co/docs/datasets/main/en/package_reference/main_classes#datasets.Dataset.remove_columns "\(in datasets vmain\)") to remove these columns, as they will be used as inputs otherwise. You can also use [`Dataset.select_columns`](https://huggingface.co/docs/datasets/main/en/package_reference/main_classes#datasets.Dataset.select_columns "\(in datasets vmain\)") to keep only the desired columns.

Pre-existing Datasetsïƒ•

The [Hugging Face Hub](https://huggingface.co/datasets) hosts 150k+ datasets, many of which can be converted for training embedding models. We are aiming to tag all Hugging Face datasets that work out of the box with Sentence Transformers with `sentence-transformers`, allowing you to easily find them by browsing to <https://huggingface.co/datasets?other=sentence-transformers>. We strongly recommend that you browse these datasets to find training datasets that might be useful for your tasks.

These are some of the popular pre-existing datasets tagged as `sentence-transformers` that can be used to train and fine-tune SentenceTransformer models:

Dataset | Description

---|---

[GooAQ](https://huggingface.co/datasets/sentence-transformers/gooaq) | (Question, Answer) pairs from Google auto suggest

[Yahoo Answers](https://huggingface.co/datasets/sentence-transformers/yahoo-answers) | (Title+Question, Answer), (Title, Answer), (Title, Question), (Question, Answer) pairs from Yahoo Answers

[MS MARCO Triplets (msmarco-distilbert-base-tas-b)](https://huggingface.co/datasets/sentence-transformers/msmarco-msmarco-distilbert-base-tas-b) | (Question, Answer, Negative) triplets from MS MARCO Passages dataset with mined negatives

[MS MARCO Triplets

(msmarco-distilbert-base-v3)](https://huggingface.co/datasets/sentence-transformers/msmarco-msmarco-distilbert-base-v3) | (Question, Answer, Negative) triplets from MS MARCO Passages dataset with mined negatives

[MS MARCO Triplets (msmarco-MiniLM-L-6-v3)](https://huggingface.co/datasets/sentence-transformers/msmarco-msmarco-MiniLM-L-6-v3) | (Question, Answer, Negative) triplets from MS MARCO Passages dataset with mined negatives

[MS MARCO Triplets (distilbert-margin-mse-cls-dot-v2)](https://huggingface.co/datasets/sentence-transformers/msmarco-distilbert-margin-mse-cls-dot-v2) | (Question, Answer, Negative) triplets from MS MARCO Passages dataset with mined negatives

[MS MARCO Triplets (distilbert-margin-mse-cls-dot-v1)](https://huggingface.co/datasets/sentence-transformers/msmarco-distilbert-margin-mse-cls-dot-v1) | (Question, Answer, Negative) triplets from MS MARCO Passages dataset with mined negatives

[MS MARCO Triplets (distilbert-margin-mse-mean-dot-v1)](https://huggingface.co/datasets/sentence-transformers/msmarco-distilbert-margin-mse-mean-dot-v1) | (Question, Answer, Negative) triplets from MS MARCO Passages dataset with mined negatives

[MS MARCO Triplets (mpnet-margin-mse-mean-v1)](https://huggingface.co/datasets/sentence-transformers/msmarco-mpnet-margin-mse-mean-v1) | (Question, Answer, Negative) triplets from MS MARCO Passages dataset with mined negatives

[MS MARCO Triplets (co-condenser-margin-mse-cls-v1)](https://huggingface.co/datasets/sentence-transformers/msmarco-co-condenser-margin-mse-cls-v1) | (Question, Answer, Negative) triplets from MS MARCO Passages dataset with mined negatives

[MS MARCO Triplets (distilbert-margin-mse-mnrl-mean-v1)](https://huggingface.co/datasets/sentence-transformers/msmarco-distilbert-margin-mse-mnrl-mean-v1) | (Question, Answer, Negative) triplets from MS MARCO Passages dataset with mined negatives

[MS MARCO Triplets (distilbert-margin-mse-sym-mnrl-mean-v1)](https://huggingface.co/datasets/sentence-transformers/msmarco-distilbert-margin-mse-sym-mnrl-mean-v1) | (Question, Answer, Negative) triplets from MS MARCO Passages dataset with mined negatives

[MS MARCO Triplets (distilbert-margin-mse-sym-mnrl-mean-v2)](https://huggingface.co/datasets/sentence-transformers/msmarco-distilbert-margin-mse-sym-mnrl-mean-v2) | (Question, Answer, Negative) triplets from MS MARCO Passages dataset with mined negatives

[MS MARCO Triplets (co-condenser-margin-mse-sym-mnrl-mean-v1)](https://huggingface.co/datasets/sentence-transformers/msmarco-co-condenser-margin-mse-sym-mnrl-mean-v1) | (Question, Answer, Negative) triplets from MS MARCO Passages dataset with mined negatives

[MS MARCO Triplets (BM25)](https://huggingface.co/datasets/sentence-transformers/msmarco-bm25) | (Question, Answer, Negative) triplets from MS MARCO Passages dataset with mined negatives

[Stack Exchange Duplicates](https://huggingface.co/datasets/sentence-transformers/stackexchange-duplicates) | (Title, Title), (Title+Body, Title+Body), (Body, Body) pairs of duplicate questions from StackExchange

[ELI5](https://huggingface.co/datasets/sentence-transformers/eli5) | (Question, Answer) pairs from ELI5 dataset

[SQuAD](https://huggingface.co/datasets/sentence-transformers/squad) | (Question, Answer) pairs from SQuAD dataset

[WikiHow](https://huggingface.co/datasets/sentence-transformers/wikihow) | (Summary, Text) pairs from WikiHow

[Amazon Reviews 2018](https://huggingface.co/datasets/sentence-transformers/amazon-reviews) | (Title, review) pairs from Amazon Reviews

[Natural Questions](https://huggingface.co/datasets/sentence-transformers/natural-questions) | (Query, Answer) pairs from the Natural Questions dataset

[Amazon QA](https://huggingface.co/datasets/sentence-transformers/amazon-qa) | (Question, Answer) pairs from Amazon

[S2ORC](https://huggingface.co/datasets/sentence-transformers/s2orc) | (Title, Abstract), (Abstract, Citation), (Title, Citation) pairs of scientific papers

[Quora Duplicates](https://huggingface.co/datasets/sentence-transformers/quora-duplicates) | Duplicate question pairs from Quora

[WikiAnswers](https://huggingface.co/datasets/sentence-transformers/wikianswers-duplicates) | Duplicate question pairs from WikiAnswers

[AGNews](https://huggingface.co/datasets/sentence-transformers/agnews) | (Title, Description) pairs of news articles from the AG News dataset

[AllNLI](https://huggingface.co/datasets/sentence-transformers/all-nli) | (Anchor, Entailment, Contradiction) triplets from SNLI + MultiNLI

[NPR](https://huggingface.co/datasets/sentence-transformers/npr) | (Title, Body) pairs from the npr.org website

[SPECTER](https://huggingface.co/datasets/sentence-transformers/specter) | (Title, Positive Title, Negative Title) triplets of Scientific Publications from Specter

[Simple Wiki](https://huggingface.co/datasets/sentence-transformers/simple-wiki) | (English, Simple English) pairs from Wikipedia

[PAQ](https://huggingface.co/datasets/sentence-transformers/paq) | (Query, Answer) from the Probably-Asked Questions dataset

[altlex](https://huggingface.co/datasets/sentence-transformers/altlex) | (English, Simple English)

pairs from Wikipedia

[CC News](https://huggingface.co/datasets/sentence-transformers/ccnews) | (Title, article) pairs from the CC News dataset

[CodeSearchNet](https://huggingface.co/datasets/sentence-transformers/codesearchnet) | (Comment, Code) pairs from open source libraries on GitHub

[Sentence Compression](https://huggingface.co/datasets/sentence-transformers/sentence-compression) | (Long text, Short text) pairs from the Sentence Compression dataset

[Trivia QA](https://huggingface.co/datasets/sentence-transformers/trivia-qa) | (Query, Answer) pairs from the TriviaQA dataset

[Flickr30k Captions](https://huggingface.co/datasets/sentence-transformers/flickr30k-captions) | Duplicate captions from the Flickr30k dataset

[xsum](https://huggingface.co/datasets/sentence-transformers/xsum) | (News Article, Summary) pairs from XSUM dataset

[Coco Captions](https://huggingface.co/datasets/sentence-transformers/coco-captions) | Duplicate captions from the Coco Captions dataset

[Parallel Sentences: Europarl](https://huggingface.co/datasets/sentence-transformers/parallel-sentences-europarl) | (English, Non-English) pairs across numerous languages

[Parallel Sentences: Global Voices](https://huggingface.co/datasets/sentence-transformers/parallel-sentences-global-voices) | (English, Non-English) pairs across numerous languages

[Parallel Sentences: MUSE](https://huggingface.co/datasets/sentence-transformers/parallel-sentences-muse) | (English, Non-English) pairs across numerous languages

[Parallel Sentences: JW300](https://huggingface.co/datasets/sentence-transformers/parallel-sentences-jw300) | (English,

Non-English) pairs across numerous languages

[Parallel Sentences: News Commentary](https://huggingface.co/datasets/sentence-transformers/parallel-sentences-news-commentary) | (English, Non-English) pairs across numerous languages

[Parallel Sentences: OpenSubtitles](https://huggingface.co/datasets/sentence-transformers/parallel-sentences-opensubtitles) | (English, Non-English) pairs across numerous languages

[Parallel Sentences: Talks](https://huggingface.co/datasets/sentence-transformers/parallel-sentences-talks) | (English, Non-English) pairs across numerous languages

[Parallel Sentences: Tatoeba](https://huggingface.co/datasets/sentence-transformers/parallel-sentences-tatoeba) | (English, Non-English) pairs across numerous languages

[Parallel Sentences: WikiMatrix](https://huggingface.co/datasets/sentence-transformers/parallel-sentences-wikimatrix) | (English, Non-English) pairs across numerous languages

[Parallel Sentences: WikiTitles](https://huggingface.co/datasets/sentence-transformers/parallel-sentences-wikititles) | (English, Non-English) pairs across numerous languages

Note

We advise users to tag datasets that can be used for training embedding models with `sentence-transformers` by adding `tags: sentence-transformers`. We would also gladly accept high quality datasets to be added to the list above for all to see and use.

[Previous](training_overview.html "Training Overview") [Next

](loss_overview.html "Loss Overview")

* * *