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

Getting Started

* [Installation](../../docs/installation.html)
  * [Install with pip](../../docs/installation.html#install-with-pip)
  * [Install with Conda](../../docs/installation.html#install-with-conda)
  * [Install from Source](../../docs/installation.html#install-from-source)
  * [Editable Install](../../docs/installation.html#editable-install)
  * [Install PyTorch with CUDA support](../../docs/installation.html#install-pytorch-with-cuda-support)
* [Quickstart](../../docs/quickstart.html)
  * [Sentence Transformer](../../docs/quickstart.html#sentence-transformer)
  * [Cross Encoder](../../docs/quickstart.html#cross-encoder)
  * [Next Steps](../../docs/quickstart.html#next-steps)

Sentence Transformer

* [Usage](../../docs/sentence_transformer/usage/usage.html)
  * [Computing Embeddings](../applications/computing-embeddings/README.html)
    * [Initializing a Sentence Transformer Model](../applications/computing-embeddings/README.html#initializing-a-sentence-transformer-model)
    * [Calculating Embeddings](../applications/computing-embeddings/README.html#calculating-embeddings)
      * [Prompt Templates](../applications/computing-embeddings/README.html#prompt-templates)
      * [Input Sequence Length](../applications/computing-embeddings/README.html#id1)
    * [Multi-Process / Multi-GPU

Encoding](../applications/computing-embeddings/README.html#multi-process-multi-gpu-encoding)

* [Semantic Textual Similarity](../../docs/sentence_transformer/usage/semantic_textual_similarity.html)

* [Similarity Calculation](../../docs/sentence_transformer/usage/semantic_textual_similarity.html#similarity-calculation)

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

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

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

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

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

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

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

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

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

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

* [Retrieve & Re-Rank](../applications/retrieve_rerank/README.html)

* [Retrieve & Re-Rank Pipeline](../applications/retrieve_rerank/README.html#retrieve-re-rank-pipeline)

* [Retrieval: Bi-Encoder](../applications/retrieve_rerank/README.html#retrieval-bi-encoder)

* [Re-Ranker: Cross-Encoder](../applications/retrieve_rerank/README.html#re-ranker-cross-encoder)

* [Example Scripts](../applications/retrieve_rerank/README.html#example-scripts)

* [Pre-trained Bi-Encoders (Retrieval)](../applications/retrieve_rerank/README.html#pre-trained-bi-encoders-retrieval)
* [Pre-trained Cross-Encoders (Re-Ranker)](../applications/retrieve_rerank/README.html#pre-trained-cross-encoders-re-ranker)
* [Clustering](../applications/clustering/README.html)
* [k-Means](../applications/clustering/README.html#k-means)
* [Agglomerative Clustering](../applications/clustering/README.html#agglomerative-clustering)
* [Fast Clustering](../applications/clustering/README.html#fast-clustering)
* [Topic Modeling](../applications/clustering/README.html#topic-modeling)
* [Paraphrase Mining](../applications/paraphrase-mining/README.html)
* [`paraphrase_mining()`](../applications/paraphrase-mining/README.html#sentence_transformers.util.paraphrase_mining)
* [Translated Sentence Mining](../applications/parallel-sentence-mining/README.html)
* [Margin Based Mining](../applications/parallel-sentence-mining/README.html#margin-based-mining)
* [Examples](../applications/parallel-sentence-mining/README.html#examples)
* [Image Search](../applications/image-search/README.html)
* [Installation](../applications/image-search/README.html#installation)
* [Usage](../applications/image-search/README.html#usage)
* [Examples](../applications/image-search/README.html#examples)
* [Embedding Quantization](../applications/embedding-quantization/README.html)
* [Binary Quantization](../applications/embedding-quantization/README.html#binary-quantization)
* [Scalar (int8) Quantization](../applications/embedding-quantization/README.html#scalar-int8-quantization)
* [Additional

extensions](../applications/embedding-quantization/README.html#additional-extensions)
    * [Demo](../applications/embedding-quantization/README.html#demo)
    * [Try it yourself](../applications/embedding-quantization/README.html#try-it-yourself)
  * [Speeding up Inference](../../docs/sentence_transformer/usage/efficiency.html)
    * [PyTorch](../../docs/sentence_transformer/usage/efficiency.html#pytorch)
    * [ONNX](../../docs/sentence_transformer/usage/efficiency.html#onnx)
    * [OpenVINO](../../docs/sentence_transformer/usage/efficiency.html#openvino)
    * [Benchmarks](../../docs/sentence_transformer/usage/efficiency.html#benchmarks)
  * [Creating Custom Models](../../docs/sentence_transformer/usage/custom_models.html)
    * [Structure of Sentence Transformer Models](../../docs/sentence_transformer/usage/custom_models.html#structure-of-sentence-transformer-models)
    * [Sentence Transformer Model from a Transformers Model](../../docs/sentence_transformer/usage/custom_models.html#sentence-transformer-model-from-a-transformers-model)
* [Pretrained Models](../../docs/sentence_transformer/pretrained_models.html)
  * [Original Models](../../docs/sentence_transformer/pretrained_models.html#original-models)
    * [Semantic Search Models](../../docs/sentence_transformer/pretrained_models.html#semantic-search-models)
    * [Multi-QA Models](../../docs/sentence_transformer/pretrained_models.html#multi-qa-models)
    * [MSMARCO Passage Models](../../docs/sentence_transformer/pretrained_models.html#msmarco-passage-models)
    * [Multilingual Models](../../docs/sentence_transformer/pretrained_models.html#multilingual-models)
    * [Semantic Similarity Models](../../docs/sentence_transformer/pretrained_models.html#semantic-similarity-models)
    * [Bitext Mining](../../docs/sentence_transformer/pretrained_models.html#bitext-mining)

* [Image & Text-Models](../../docs/sentence_transformer/pretrained_models.html#image-text-models)
* [INSTRUCTOR models](../../docs/sentence_transformer/pretrained_models.html#instructor-models)
* [Scientific Similarity Models](../../docs/sentence_transformer/pretrained_models.html#scientific-similarity-models)
* [Training Overview](../../docs/sentence_transformer/training_overview.html)
* [Why Finetune?](../../docs/sentence_transformer/training_overview.html#why-finetune)
* [Training Components](../../docs/sentence_transformer/training_overview.html#training-components)
* [Dataset](../../docs/sentence_transformer/training_overview.html#dataset)
* [Dataset Format](../../docs/sentence_transformer/training_overview.html#dataset-format)
* [Loss Function](../../docs/sentence_transformer/training_overview.html#loss-function)
* [Training Arguments](../../docs/sentence_transformer/training_overview.html#training-arguments)
* [Evaluator](../../docs/sentence_transformer/training_overview.html#evaluator)
* [Trainer](../../docs/sentence_transformer/training_overview.html#trainer)
* [Callbacks](../../docs/sentence_transformer/training_overview.html#callbacks)
* [Multi-Dataset Training](../../docs/sentence_transformer/training_overview.html#multi-dataset-training)
* [Deprecated Training](../../docs/sentence_transformer/training_overview.html#deprecated-training)
* [Best Base Embedding Models](../../docs/sentence_transformer/training_overview.html#best-base-embedding-models)
* [Dataset Overview](../../docs/sentence_transformer/dataset_overview.html)
* [Datasets on the Hugging Face Hub](../../docs/sentence_transformer/dataset_overview.html#datasets-on-the-hugging-face-hub)

* [Pre-existing Datasets](../../docs/sentence_transformer/dataset_overview.html#pre-existing-datasets)
* [Loss Overview](../../docs/sentence_transformer/loss_overview.html)
* [Loss modifiers](../../docs/sentence_transformer/loss_overview.html#loss-modifiers)
* [Distillation](../../docs/sentence_transformer/loss_overview.html#distillation)
* [Commonly used Loss Functions](../../docs/sentence_transformer/loss_overview.html#commonly-used-loss-functions)
* [Custom Loss Functions](../../docs/sentence_transformer/loss_overview.html#custom-loss-functions)
* [Training Examples](../../docs/sentence_transformer/training/examples.html)
* [Semantic Textual Similarity](../training/sts/README.html)
* [Training data](../training/sts/README.html#training-data)
* [Loss Function](../training/sts/README.html#loss-function)
* [Natural Language Inference](../training/nli/README.html)
* [Data](../training/nli/README.html#data)
* [SoftmaxLoss](../training/nli/README.html#softmaxloss)
* [MultipleNegativesRankingLoss](../training/nli/README.html#multiplenegativesrankingloss)
* [Paraphrase Data](../training/paraphrases/README.html)
* [Pre-Trained Models](../training/paraphrases/README.html#pre-trained-models)
* [Quora Duplicate Questions](../training/quora_duplicate_questions/README.html)
* [Training](../training/quora_duplicate_questions/README.html#training)
* [MultipleNegativesRankingLoss](../training/quora_duplicate_questions/README.html#multiplenegativesrankingloss)
* [Pretrained Models](../training/quora_duplicate_questions/README.html#pretrained-models)
* [MS MARCO](../training/ms_marco/README.html)
* [Bi-Encoder](../training/ms_marco/README.html#bi-encoder)

* [Matryoshka Embeddings](../training/matryoshka/README.html)
  * [Use Cases](../training/matryoshka/README.html#use-cases)
  * [Results](../training/matryoshka/README.html#results)
  * [Training](../training/matryoshka/README.html#training)
  * [Inference](../training/matryoshka/README.html#inference)
  * [Code Examples](../training/matryoshka/README.html#code-examples)
* [Adaptive Layers](../training/adaptive_layer/README.html)
  * [Use Cases](../training/adaptive_layer/README.html#use-cases)
  * [Results](../training/adaptive_layer/README.html#results)
  * [Training](../training/adaptive_layer/README.html#training)
  * [Inference](../training/adaptive_layer/README.html#inference)
  * [Code Examples](../training/adaptive_layer/README.html#code-examples)
* [Multilingual Models](../training/multilingual/README.html)
  * [Extend your own models](../training/multilingual/README.html#extend-your-own-models)
  * [Training](../training/multilingual/README.html#training)
  * [Datasets](../training/multilingual/README.html#datasets)
  * [Sources for Training Data](../training/multilingual/README.html#sources-for-training-data)
  * [Evaluation](../training/multilingual/README.html#evaluation)
  * [Available Pre-trained Models](../training/multilingual/README.html#available-pre-trained-models)
  * [Usage](../training/multilingual/README.html#usage)
  * [Performance](../training/multilingual/README.html#performance)
  * [Citation](../training/multilingual/README.html#citation)
* [Model Distillation](../training/distillation/README.html)
  * [Knowledge Distillation](../training/distillation/README.html#knowledge-distillation)
  * [Speed - Performance Trade-Off](../training/distillation/README.html#speed-performance-trade-off)

* [Dimensionality Reduction](../training/distillation/README.html#dimensionality-reduction)
* [Quantization](../training/distillation/README.html#quantization)
* [Augmented SBERT](../training/data_augmentation/README.html)
* [Motivation](../training/data_augmentation/README.html#motivation)
* [Extend to your own datasets](../training/data_augmentation/README.html#extend-to-your-own-datasets)
* [Methodology](../training/data_augmentation/README.html#methodology)
* [Scenario 1: Limited or small annotated datasets (few labeled sentence-pairs)](../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)](../training/data_augmentation/README.html#scenario-2-no-annotated-datasets-only-unlabeled-sentence-pairs)
* [Training](../training/data_augmentation/README.html#training)
* [Citation](../training/data_augmentation/README.html#citation)
* [Training with Prompts](../training/prompts/README.html)
* [What are Prompts?](../training/prompts/README.html#what-are-prompts)
* [Why would we train with Prompts?](../training/prompts/README.html#why-would-we-train-with-prompts)
* [How do we train with Prompts?](../training/prompts/README.html#how-do-we-train-with-prompts)
* [Training with PEFT Adapters](../training/peft/README.html)
* [Compatibility Methods](../training/peft/README.html#compatibility-methods)
* [Adding a New Adapter](../training/peft/README.html#adding-a-new-adapter)
* [Loading a Pretrained Adapter](../training/peft/README.html#loading-a-pretrained-adapter)
* [Training Script](../training/peft/README.html#training-script)
* Unsupervised Learning

      * TSDAE

      * SimCSE

      * CT

      * CT (In-Batch Negative Sampling)

      * Masked Language Model (MLM)

      * GenQ

      * GPL

      * Performance Comparison

    * [Domain Adaptation](../domain_adaptation/README.html)

      * [Domain Adaptation vs. Unsupervised Learning](../domain_adaptation/README.html#domain-adaptation-vs-unsupervised-learning)

      * [Adaptive Pre-Training](../domain_adaptation/README.html#adaptive-pre-training)

      * [GPL: Generative Pseudo-Labeling](../domain_adaptation/README.html#gpl-generative-pseudo-labeling)

    * [Hyperparameter Optimization](../training/hpo/README.html)

      * [HPO Components](../training/hpo/README.html#hpo-components)

      * [Putting It All Together](../training/hpo/README.html#putting-it-all-together)

      * [Example Scripts](../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](../applications/retrieve_rerank/README.html)

      * [Retrieve & Re-Rank

Pipeline](../applications/retrieve_rerank/README.html#retrieve-re-rank-pipeline)
* [Retrieval: Bi-Encoder](../applications/retrieve_rerank/README.html#retrieval-bi-encoder)
* [Re-Ranker: Cross-Encoder](../applications/retrieve_rerank/README.html#re-ranker-cross-encoder)
* [Example Scripts](../applications/retrieve_rerank/README.html#example-scripts)
* [Pre-trained Bi-Encoders (Retrieval)](../applications/retrieve_rerank/README.html#pre-trained-bi-encoders-retrieval)
* [Pre-trained Cross-Encoders (Re-Ranker)](../applications/retrieve_rerank/README.html#pre-trained-cross-encoders-re-ranker)
* [Pretrained Models](../../docs/cross_encoder/pretrained_models.html)
* [MS MARCO](../../docs/cross_encoder/pretrained_models.html#ms-marco)
* [SQuAD (QNLI)](../../docs/cross_encoder/pretrained_models.html#squad-qnli)
* [STSbenchmark](../../docs/cross_encoder/pretrained_models.html#stsbenchmark)
* [Quora Duplicate Questions](../../docs/cross_encoder/pretrained_models.html#quora-duplicate-questions)
* [NLI](../../docs/cross_encoder/pretrained_models.html#nli)
* [Community Models](../../docs/cross_encoder/pretrained_models.html#community-models)
* [Training Overview](../../docs/cross_encoder/training_overview.html)
* [Training Examples](../../docs/cross_encoder/training/examples.html)
* [MS MARCO](../training/ms_marco/cross_encoder_README.html)
* [Cross-Encoder](../training/ms_marco/cross_encoder_README.html#cross-encoder)
* [Cross-Encoder Knowledge Distillation](../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#similarityfunction)
* [Trainer](../../docs/package_reference/sentence_transformer/trainer.html)
*
[SentenceTransformerTrainer](../../docs/package_reference/sentence_transformer/trainer.html#sentencetransformertrainer)
* [Training Arguments](../../docs/package_reference/sentence_transformer/training_args.html)
*
[SentenceTransformerTrainingArguments](../../docs/package_reference/sentence_transformer/training_args.html#sentencetransformertrainingarguments)
* [Losses](../../docs/package_reference/sentence_transformer/losses.html)
*
[BatchAllTripletLoss](../../docs/package_reference/sentence_transformer/losses.html#batchalltripletloss)
*
[BatchHardSoftMarginTripletLoss](../../docs/package_reference/sentence_transformer/losses.html#b

atchhardsoftmargintripletloss)

*

[BatchHardTripletLoss](../../docs/package_reference/sentence_transformer/losses.html#batchhardtripletloss)

*

[BatchSemiHardTripletLoss](../../docs/package_reference/sentence_transformer/losses.html#batchsemihardtripletloss)

*

[ContrastiveLoss](../../docs/package_reference/sentence_transformer/losses.html#contrastiveloss)

*

[OnlineContrastiveLoss](../../docs/package_reference/sentence_transformer/losses.html#onlinecontrastiveloss)

*

[ContrastiveTensionLoss](../../docs/package_reference/sentence_transformer/losses.html#contrastivetensionloss)

*

[ContrastiveTensionLossInBatchNegatives](../../docs/package_reference/sentence_transformer/losses.html#contrastivetensionlossinbatchnegatives)

    * [CoSENTLoss](../../docs/package_reference/sentence_transformer/losses.html#cosentloss)
    * [AnglELoss](../../docs/package_reference/sentence_transformer/losses.html#angleloss)

*

[CosineSimilarityLoss](../../docs/package_reference/sentence_transformer/losses.html#cosinesimilarityloss)

*

[DenoisingAutoEncoderLoss](../../docs/package_reference/sentence_transformer/losses.html#denoisingautoencoderloss)

*

[GISTEmbedLoss](../../docs/package_reference/sentence_transformer/losses.html#gistembedloss)
*
[CachedGISTEmbedLoss](../../docs/package_reference/sentence_transformer/losses.html#cachedgistembedloss)
    * [MSELoss](../../docs/package_reference/sentence_transformer/losses.html#mseloss)
*
[MarginMSELoss](../../docs/package_reference/sentence_transformer/losses.html#marginmseloss)
*
[MatryoshkaLoss](../../docs/package_reference/sentence_transformer/losses.html#matryoshkaloss)
*
[Matryoshka2dLoss](../../docs/package_reference/sentence_transformer/losses.html#matryoshka2dloss)
*
[AdaptiveLayerLoss](../../docs/package_reference/sentence_transformer/losses.html#adaptivelayerloss)
*
[MegaBatchMarginLoss](../../docs/package_reference/sentence_transformer/losses.html#megabatchmarginloss)
*
[MultipleNegativesRankingLoss](../../docs/package_reference/sentence_transformer/losses.html#multiplenegativesrankingloss)
*
[CachedMultipleNegativesRankingLoss](../../docs/package_reference/sentence_transformer/losses.html#cachedmultiplenegativesrankingloss)
*
[MultipleNegativesSymmetricRankingLoss](../../docs/package_reference/sentence_transformer/losses.html#multiplenegativessymmetricrankingloss)

*
[CachedMultipleNegativesSymmetricRankingLoss](../../docs/package_reference/sentence_transformer/losses.html#cachedmultiplenegativessymmetricrankingloss)
    * [SoftmaxLoss](../../docs/package_reference/sentence_transformer/losses.html#softmaxloss)
    * [TripletLoss](../../docs/package_reference/sentence_transformer/losses.html#tripletloss)
  * [Samplers](../../docs/package_reference/sentence_transformer/sampler.html)
*
[BatchSamplers](../../docs/package_reference/sentence_transformer/sampler.html#batchsamplers)
*
[MultiDatasetBatchSamplers](../../docs/package_reference/sentence_transformer/sampler.html#multidatasetbatchsamplers)
  * [Evaluation](../../docs/package_reference/sentence_transformer/evaluation.html)
*
[BinaryClassificationEvaluator](../../docs/package_reference/sentence_transformer/evaluation.html#binaryclassificationevaluator)
*
[EmbeddingSimilarityEvaluator](../../docs/package_reference/sentence_transformer/evaluation.html#embeddingsimilarityevaluator)
*
[InformationRetrievalEvaluator](../../docs/package_reference/sentence_transformer/evaluation.html#informationretrievalevaluator)
*
[NanoBEIREvaluator](../../docs/package_reference/sentence_transformer/evaluation.html#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#nodup
licatesdataloader)
* [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#se
ntence_transformers.quantization.quantize_embeddings)
*
[`semantic_search_faiss()`](../../docs/package_reference/sentence_transformer/quantization.html#s
entence_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#cebinary
accuracyevaluator)
*

[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_si

m)

*

[`pairwise_cos_sim()`](../../docs/package_reference/util.html#sentence_transformers.util.pairwise_cos_sim)

*

[`pairwise_dot_score()`](../../docs/package_reference/util.html#sentence_transformers.util.pairwise_dot_score)

*

[`pairwise_euclidean_sim()`](../../docs/package_reference/util.html#sentence_transformers.util.pairwise_euclidean_sim)

*

[`pairwise_manhattan_sim()`](../../docs/package_reference/util.html#sentence_transformers.util.pairwise_manhattan_sim)

__[Sentence Transformers](../../index.html)

* [](../../index.html)
* [Training Examples](../../docs/sentence_transformer/training/examples.html)
* Unsupervised Learning

* [Edit on GitHub](https://github.com/UKPLab/sentence-transformers/blob/master/examples/unsupervised_learning/README.md)

* * *

Unsupervised Learningïƒ•

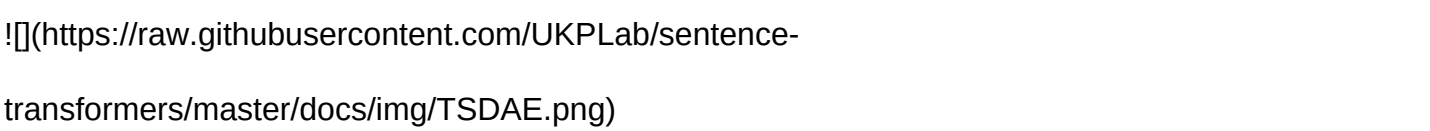
This page contains a collection of unsupervised learning methods to learn sentence embeddings. The methods have in common that they **do not require labeled training data**. Instead, they can learn semantically meaningful sentence embeddings just from the text itself.

Note

Unsupervised learning approaches are still an activate research area and in many cases the models perform rather poorly compared to models that are using training pairs as provided in our [training data collection](https://huggingface.co/collections/sentence-transformers/embedding-model-datasets-6644d7a3673a511914aa7552). A better approach is [Domain Adaptation](../domain_adaptation/README.html) where you combine unsupervised learning on your target domain with existent labeled data. This should give the best performance on your specific corpus.

TSDAEïƒ•

In our work [TSDAE (Transformer-based Denoising AutoEncoder)](https://arxiv.org/abs/2104.06979) we present an unsupervised sentence embedding learning method based on denoising auto-encoders:



We add noise to the input text, in our case, we delete about 60% of the words in the text. The encoder maps this input to a fixed-sized sentence embeddings.

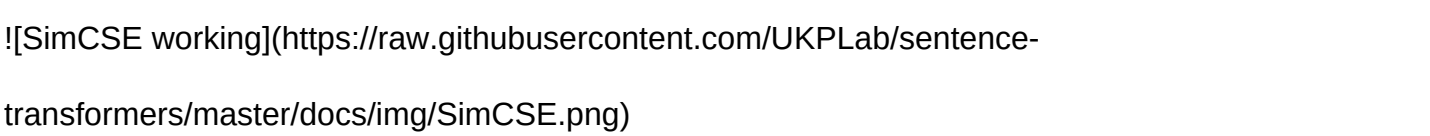
A decoder then tries to re-create the original text without the noise. Later, we use the encoder as the sentence embedding methods.

See **[TSDAE](TSDAE/README.html)** for more information and training examples.

SimCSEïƒ•

Gao et al. present in [SimCSE: Simple Contrastive Learning of Sentence Embeddings](https://arxiv.org/abs/2104.08821) a method that passes the same sentence twice to the sentence embedding encoder. Due to the drop-out, it will be encoded at slightly different positions in vector space.

The distance between these two embeddings will be minimized, while the distance to other embeddings of the other sentences in the same batch will be maximized.

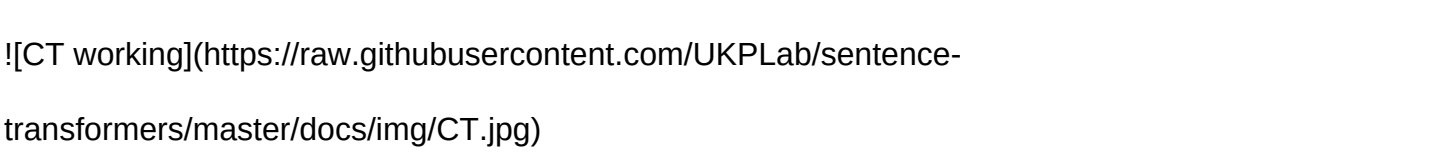


See **[SimCSE](SimCSE/README.html)** for more information and training examples.

CTïƒ•

Carlsson et al. present in [Semantic Re-Tuning With Contrastive Tension (CT)](https://openreview.net/pdf?id=Ov_sMNau-PF) an unsupervised method that uses two models: If the same sentences are passed to Model1 and Model2, then

the respective sentence embeddings should get a large dot-score. If the different sentences are passed, then the sentence embeddings should get a low score.



See **[CT](CT/README.html)** for more information and training examples.

CT (In-Batch Negative Sampling)ïƒ•

The CT method from Carlsson et al. provides sentence pairs to the two models. This can be improved by using in-batch negative sampling: Model1 and Model2 both encode the same set of sentences. We maximize the scores for matching indexes (i.e. Model1(S_i) and Model2(S_i)) while we minimize the scores for different indexes (i.e. Model1(S_i) and Model2(S_j) for i != j).

See **[CT_In-Batch_Negatives](CT_In-Batch_Negatives/README.html)** for more information and training examples.

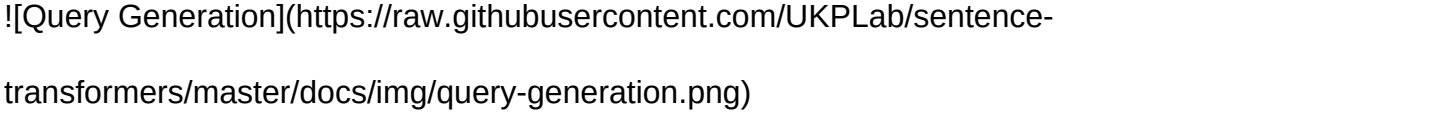
Masked Language Model (MLM)ïƒ•

BERT showed that Masked Language Model (MLM) is a powerful pre-training approach. It is advisable to first run MLM a large dataset from your domain before you do fine-tuning. See **[MLM](MLM/README.html)** for more information and training examples.

GenQï ƒ •

In our paper [BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models](https://arxiv.org/abs/2104.08663) we present a method to learn a semantic search method by generating queries for given passages. This method has been improved in [GPL: Generative Pseudo Labeling for Unsupervised Domain Adaptation of Dense Retrieval](https://arxiv.org/abs/2112.07577).

We pass all passages in our collection through a trained T5 model, which generates potential queries from users. We then use these (query, passage) pairs to train a SentenceTransformer model.

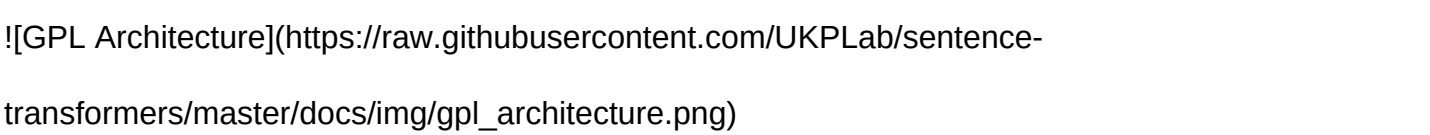


See **[GenQ](query_generation/README.html)** for more information and training examples. See **[GPL](../domain_adaptation/README.html)** for the improved version that uses a multi-step training approach.

GPLï ƒ •

In [GPL: Generative Pseudo Labeling for Unsupervised Domain Adaptation of Dense Retrieval](https://arxiv.org/abs/2112.07577) we show an improved version of GenQ, which combines the generation with negative mining and pseudo labeling using a Cross-Encoder. It leads to significantly improved results.
See **[Domain Adaptation](../domain_adaptation/README.html)** for more

information.



Performance Comparisonïƒ•

In our paper [TSDAE](https://arxiv.org/abs/2104.06979) we compare approaches for sentence embedding tasks, and in [GPL](https://arxiv.org/abs/2112.07577) we compare them for semantic search tasks (given a query, find relevant passages). While the unsupervised approach achieve acceptable performances for sentence embedding tasks, they perform poorly for semantic search tasks.

[Previous](../training/peft/README.html "Training with PEFT Adapters") [Next](../domain_adaptation/README.html "Domain Adaptation")

* * *

(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).