

[! (../_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 \(Only unlabeled sentence-pairs\)\]\(../training/data_augmentation/README.html#scenario-2-no-annotated-datasets-only-unlabeled-sentence-pairs\)](#)
- * [\[Training\]\(../training/data_augmentation/README.html#training\)](#)
- * [\[Citation\]\(../training/data_augmentation/README.html#citation\)](#)
- * [\[Training with Prompts\]\(../training/prompts/README.html\)](#)
- * [\[What are Prompts?\]\(../training/prompts/README.html#what-are-prompts\)](#)
 - * [\[Why would we train with Prompts?\]\(../training/prompts/README.html#why-would-we-train-with-prompts\)](#)
 - * [\[How do we train with Prompts?\]\(../training/prompts/README.html#how-do-we-train-with-prompts\)](#)
- * [\[Training with PEFT Adapters\]\(../training/peft/README.html\)](#)
- * [\[Compatibility Methods\]\(../training/peft/README.html#compatibility-methods\)](#)
- * [\[Adding a New Adapter\]\(../training/peft/README.html#adding-a-new-adapter\)](#)
- * [\[Loading a Pretrained Adapter\]\(../training/peft/README.html#loading-a-pretrained-adapter\)](#)
- * [\[Training Script\]\(../training/peft/README.html#training-script\)](#)
- * [\[Unsupervised Learning\]\(../unsupervised_learning/README.html\)](#)

* [TSDAE](../unsupervised_learning/README.html#tsdae)

* [SimCSE](../unsupervised_learning/README.html#simcse)

* [CT](../unsupervised_learning/README.html#ct)

* [CT (In-Batch Negative Sampling)](../unsupervised_learning/README.html#ct-in-batch-negative-sampling)

* [Masked Language Model (MLM)](../unsupervised_learning/README.html#masked-language-model-mlm)

* [GenQ](../unsupervised_learning/README.html#genq)

* [GPL](../unsupervised_learning/README.html#gpl)

* [Performance Comparison](../unsupervised_learning/README.html#performance-comparison)

* Domain Adaptation

* Domain Adaptation vs. Unsupervised Learning

* Adaptive Pre-Training

* 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#batchsemi-hardtripletloss)

*

[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)

* [AngleELoss](../../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#multi
datasetbatchsamplers)

* [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#nanobeir
evaluator)

*

[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_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)

* Domain Adaptation

*

[

Edit

on

GitHub](https://github.com/UKPLab/sentence-transformers/blob/master/examples/domain_adaptation/README.md)

* * *

Domain Adaptation

The goal of **Domain Adaptation** is to adapt text embedding models to your specific text domain without the need to have labeled training data.

Domain adaptation is still an active research field and there exists no perfect solution yet. However, in our two recent papers [TSDAE](<https://arxiv.org/abs/2104.06979>) and [GPL](<https://arxiv.org/abs/2112.07577>) we evaluated several methods how text embeddings model can be adapted to your specific domain. You can find an overview of these methods in my [talk on unsupervised domain adaptation](<https://youtu.be/xbdLowiQTlk>).

Domain Adaptation vs. Unsupervised Learning

There exists methods for [unsupervised text embedding learning]([./unsupervised_learning/README.html](https://github.com/unsupervised-learning/README.html)), however, they generally perform rather badly: They are not really able to learn domain specific concepts.

A much better approach is domain adaptation: Here you have an unlabeled corpus from your specific domain together with an existing labeled corpus. You can find many suitable labeled training datasets here: [Embedding Model Datasets Collection](<https://huggingface.co/collections/sentence-transformers/embedding-model-datasets-6644d7a3673a511914aa7552>)

Adaptive Pre-Training

When using adaptive pre-training, you first pre-train on your target corpus

using e.g. [Masked Language Modeling](../unsupervised_learning/MLM/README.html) or [TSDAE](../unsupervised_learning/TSDAE/README.html) and then you fine-tune on an existing training dataset (see [Embedding Model Datasets Collection](https://huggingface.co/collections/sentence-transformers/embedding-model-datasets-6644d7a3673a511914aa7552)).

! [Adaptive Pre-Training](https://raw.githubusercontent.com/UKPLab/sentence-transformers/master/docs/img/adaptive_pre-training.png)

In our paper [TSDAE](https://arxiv.org/abs/2104.06979) we evaluated several methods for domain adaptation on 4 domain specific sentence embedding tasks:

Approach	AskUbuntu	CQADupStack	Twitter	SciDocs	Avg
----------	-----------	-------------	---------	---------	-----

---	---	---	---	---	---
-----	-----	-----	-----	-----	-----

Zero-Shot Model	54.5	12.9	72.2	69.4	52.3
-----------------	------	------	------	------	------

TSDAE	59.4	**14.4**	**74.5**	**77.6**	**56.5**
-------	------	-----------------	-----------------	-----------------	-----------------

MLM	**60.6**	14.3	71.8	76.9	55.9
-----	-----------------	------	------	------	------

CT	56.4	13.4	72.4	69.7	53.0
----	------	------	------	------	------

SimCSE	56.2	13.1	71.4	68.9	52.4
--------	------	------	------	------	------

As we can see, the performance can improve up-to 8 points when you first perform pre-training on your specific corpus and then fine-tune on provided labeled training data.

In [GPL](https://arxiv.org/abs/2112.07577) we evaluate these methods for semantic search: Given a short query, find the relevant passage. Here,

performance can improve up to 10 points:

Approach | FiQA | SciFact | BioASQ | TREC-COVID | CQADupStack | Robust04 | Avg

---|---|---|---|---|---|---|---

Zero-Shot Model | 26.7 | 57.1 | 52.9 | 66.1 | 29.6 | 39.0 | 45.2

TSDAE | 29.3 | **62.8** | **55.5** | **76.1** | **31.8** | **39.4** | **49.2**

MLM | **30.2** | 60.0 | 51.3 | 69.5 | 30.4 | 38.8 | 46.7

ICT | 27.0 | 58.3 | 55.3 | 69.7 | 31.3 | 37.4 | 46.5

SimCSE | 26.7 | 55.0 | 53.2 | 68.3 | 29.0 | 37.9 | 45.0

CD | 27.0 | 62.7 | 47.7 | 65.4 | 30.6 | 34.5 | 44.7

CT | 28.3 | 55.6 | 49.9 | 63.8 | 30.5 | 35.9 | 44.0

A big **disadvantage of adaptive pre-training** is the high computational overhead, as you must first run pre-training on your corpus and then supervised learning on a labeled training dataset. The labeled trained datasets can be quite large (e.g. the ``all-*v1`` models had been trained on over 1 billion training pairs).

GPL: Generative Pseudo-Labeling

[GPL](https://arxiv.org/abs/2112.07577) overcomes the aforementioned issue: It can be applied on-top of a fine-tuned model. Hence, you can use one of the [pre-trained models](../docs/sentence_transformer/pretrained_models.html) and adapt it to your specific domain:

![GPL_Overview](https://raw.githubusercontent.com/UKPLab/sentence-transformers/master/docs/img/gpl_overview.png)

The longer you train, the better your model gets. In our experiments, we were training the models for about 1 day on a V100-GPU. GPL can be combined with adaptive pre-training, which can give another performance boost.

![[GPL_Steps]](https://raw.githubusercontent.com/UKPLab/sentence-transformers/master/docs/img/gpl_steps.png)

GPL Steps

GPL works in three phases:

![[GPL Architecture]](https://raw.githubusercontent.com/UKPLab/sentence-transformers/master/docs/img/gpl_architecture.png)

Query Generation : For a given text from our domain, we first use a T5 model that generates a possible query for the given text. E.g. when your text is *“Python is a high-level general-purpose programming language”*, the model might generate a query like *“What is Python”*. You can find various query generators on our [doc2query-hub](<https://huggingface.co/doc2query>).

Negative Mining : Next, for the generated query *“What is Python”* we mine negative passages from our corpus, i.e. passages that are similar to the query but which a user would not consider relevant. Such a negative passage could be *“Java is a high-level, class-based, object-oriented programming language”*. We do this mining using dense retrieval, i.e. we use one of the existing text embedding models and retrieve relevant paragraphs for the given query.

Pseudo Labeling : It might be that in the negative mining step we retrieve a passage that is

actually relevant for the query (like another definition for “What is Python”). To overcome this issue, we use a [Cross-Encoder](../applications/cross-encoder/README.html) to score all (query, passage)-pairs.

Training : Once we have the triplets (generated query, positive passage, mined negative passage) and the Cross-Encoder scores for (query, positive) and (query, negative) we can start training the text embedding model using [MarginMSELoss](../docs/package_reference/sentence_transformer/losses.html#marginmseloss).

The **pseudo labeling** step is quite important and which results in the increased performance compared to the previous method QGen, which treated passages just as positive (1) or negative (0). As we see in the following picture, for a generate query (“what is futures contract”), the negative mining step retrieves passages that are partly or highly relevant to the generated query. Using MarginMSELoss and the Cross-Encoder, we can identify these passages and teach the text embedding model that these passages are also relevant for the given query.

[GPL Architecture](https://raw.githubusercontent.com/UKPLab/sentence-transformers/master/docs/img/gpl_negatives.jpg)

The following tables gives an overview of GPL in comparison to adaptive pre-training (MLM and TSDAE). As mentioned, GPL can be combined with adaptive pre-training.

Approach	FiQA	SciFact	BioASQ	TREC-COVID	CQADupStack	Robust04	Avg
---	---	---	---	---	---	---	---

Zero-Shot model | 26.7 | 57.1 | 52.9 | 66.1 | 29.6 | 39.0 | 45.2

TSDAE + GPL | ****33.3**** | ****67.3**** | ****62.8**** | 74.0 | ****35.1**** | ****42.1**** | ****52.4****

GPL | 33.1 | 65.2 | 61.6 | 71.7 | 34.4 | ****42.1**** | 51.4

TSDAE | 29.3 | 62.8 | 55.5 | ****76.1**** | 31.8 | 39.4 | 49.2

MLM | 30.2 | 60.0 | 51.3 | 69.5 | 30.4 | 38.8 | 46.7

GPL Code

You can find the code for GPL here: <<https://github.com/UKPLab/gpl>>

We made the code simple to use, so that you just need to pass your corpus and everything else is handled by the training code.

Citation

If you find these resources helpful, feel free to cite our papers.

[TSDAE: Using Transformer-based Sequential Denoising Auto-Encoder for Unsupervised Sentence Embedding Learning](<https://arxiv.org/abs/2104.06979>)

```
@inproceedings{wang-2021-TSDAE,  
  title = "TSDAE: Using Transformer-based Sequential Denoising Auto-Encoder for Unsupervised Sentence Embedding Learning",  
  author = "Wang, Kexin and Reimers, Nils and Gurevych, Iryna",  
  booktitle = "Findings of the Association for Computational Linguistics: EMNLP 2021",
```

```
month = nov,  
year = "2021",  
address = "Punta Cana, Dominican Republic",  
publisher = "Association for Computational Linguistics",  
pages = "671--688",  
url = "https://arxiv.org/abs/2104.06979",  
}
```

[GPL: Generative Pseudo Labeling for Unsupervised Domain Adaptation of Dense Retrieval](<https://arxiv.org/abs/2112.07577>):

```
@inproceedings{wang-2021-GPL,  
  title = "GPL: Generative Pseudo Labeling for Unsupervised Domain Adaptation of Dense Retrieval",  
  author = "Wang, Kexin and Thakur, Nandan and Reimers, Nils and Gurevych, Iryna",  
  journal= "arXiv preprint arXiv:2112.07577",  
  month = "12",  
  year = "2021",  
  url = "https://arxiv.org/abs/2112.07577",  
}
```

[Previous](../unsupervised_learning/README.html "Unsupervised Learning")

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

* * *

(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>).