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

* [Input Sequence Length](../../../examples/applications/computing-embeddings/README.html#id1)
* [Multi-Process / Multi-GPU Encoding](../../../examples/applications/computing-embeddings/README.html#multi-process-multi-gpu-encoding)
* [Semantic Textual Similarity](../../sentence_transformer/usage/semantic_textual_similarity.html)
* [Similarity Calculation](../../sentence_transformer/usage/semantic_textual_similarity.html#similarity-calculation)
* [Semantic Search](../../../examples/applications/semantic-search/README.html)
* [Background](../../../examples/applications/semantic-search/README.html#background)
* [Symmetric vs. Asymmetric Semantic Search](../../../examples/applications/semantic-search/README.html#symmetric-vs-asymmetric-semantic-search)
* [Manual Implementation](../../../examples/applications/semantic-search/README.html#manual-implementation)
* [Optimized Implementation](../../../examples/applications/semantic-search/README.html#optimized-implementation)
* [Speed Optimization](../../../examples/applications/semantic-search/README.html#speed-optimization)
* [Elasticsearch](../../../examples/applications/semantic-search/README.html#elasticsearch)
* [Approximate Nearest Neighbor](../../../examples/applications/semantic-search/README.html#approximate-nearest-neighbor)
* [Retrieve & Re-Rank](../../../examples/applications/semantic-search/README.html#retrieve-re-rank)

* [Examples](../../../examples/applications/semantic-search/README.html#examples)
* [Retrieve & Re-Rank](../../../examples/applications/retrieve_rerank/README.html)
* [Retrieve & Re-Rank Pipeline](../../../examples/applications/retrieve_rerank/README.html#retrieve-re-rank-pipeline)
* [Retrieval: Bi-Encoder](../../../examples/applications/retrieve_rerank/README.html#retrieval-bi-encoder)
* [Re-Ranker: Cross-Encoder](../../../examples/applications/retrieve_rerank/README.html#re-ranker-cross-encoder)
* [Example Scripts](../../../examples/applications/retrieve_rerank/README.html#example-scripts)
* [Pre-trained Bi-Encoders (Retrieval)](../../../examples/applications/retrieve_rerank/README.html#pre-trained-bi-encoders-retrieval)
* [Pre-trained Cross-Encoders (Re-Ranker)](../../../examples/applications/retrieve_rerank/README.html#pre-trained-cross-encoders-re-ranker)
* [Clustering](../../../examples/applications/clustering/README.html)
* [k-Means](../../../examples/applications/clustering/README.html#k-means)
* [Agglomerative Clustering](../../../examples/applications/clustering/README.html#agglomerative-clustering)
* [Fast Clustering](../../../examples/applications/clustering/README.html#fast-clustering)
* [Topic Modeling](../../../examples/applications/clustering/README.html#topic-modeling)
* [Paraphrase Mining](../../../examples/applications/paraphrase-mining/README.html)
* [`paraphrase_mining()`](../../../examples/applications/paraphrase-mining/README.html#sentence_transformers.util.paraphrase_mining)
* [Translated Sentence

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

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

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

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

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

* [Usage](../../../examples/applications/image-search/README.html#usage)

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

* [Embedding Quantization](../../../examples/applications/embedding-quantization/README.html)

* [Binary Quantization](../../../examples/applications/embedding-quantization/README.html#binary-quantization)

* [Scalar (int8) Quantization](../../../examples/applications/embedding-quantization/README.html#scalar-int8-quantization)

* [Additional extensions](../../../examples/applications/embedding-quantization/README.html#additional-extensions)

* [Demo](../../../examples/applications/embedding-quantization/README.html#demo)

* [Try it yourself](../../../examples/applications/embedding-quantization/README.html#try-it-yourself)

* [Speeding up Inference](../../sentence_transformer/usage/efficiency.html)

* [PyTorch](../../sentence_transformer/usage/efficiency.html#pytorch)

* [ONNX](../../sentence_transformer/usage/efficiency.html#onnx)

* [OpenVINO](../../sentence_transformer/usage/efficiency.html#openvino)

* [Benchmarks](../../sentence_transformer/usage/efficiency.html#benchmarks)

* [Creating Custom Models](../../sentence_transformer/usage/custom_models.html)

gloss)

* [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)

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](../../sentence_transformer/training/distributed.html)

* [Comparison](../../sentence_transformer/training/distributed.html#comparison)

* [FSDP](../../sentence_transformer/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

    * [DenoisingAutoEncoderLoss](losses.html#denoisingautoencoderloss)
    * [GISTEmbedLoss](losses.html#gistembedloss)
    * [CachedGISTEmbedLoss](losses.html#cachedgistembedloss)
    * [MSELoss](losses.html#mseloss)
    * [MarginMSELoss](losses.html#marginmseloss)
    * [MatryoshkaLoss](losses.html#matryoshkaloss)
    * [Matryoshka2dLoss](losses.html#matryoshka2dloss)
    * [AdaptiveLayerLoss](losses.html#adaptivelayerloss)
    * [MegaBatchMarginLoss](losses.html#megabatchmarginloss)
    * [MultipleNegativesRankingLoss](losses.html#multiplenegativesrankingloss)
    * [CachedMultipleNegativesRankingLoss](losses.html#cachedmultiplenegativesrankingloss)
    * [MultipleNegativesSymmetricRankingLoss](losses.html#multiplenegativessymmetricrankingloss)
    * [CachedMultipleNegativesSymmetricRankingLoss](losses.html#cachedmultiplenegativessymmetricrankingloss)
    * [SoftmaxLoss](losses.html#softmaxloss)
    * [TripletLoss](losses.html#tripletloss)
  * [Samplers](sampler.html)
    * [BatchSamplers](sampler.html#batchsamplers)
    * [MultiDatasetBatchSamplers](sampler.html#multidatasetbatchsamplers)
  * [Evaluation](evaluation.html)
    * [BinaryClassificationEvaluator](evaluation.html#binaryclassificationevaluator)
    * [EmbeddingSimilarityEvaluator](evaluation.html#embeddingsimilarityevaluator)
    * [InformationRetrievalEvaluator](evaluation.html#informationretrievalevaluator)
    * [NanoBEIREvaluator](evaluation.html#nanobeirevaluator)
    * [MSEEvaluator](evaluation.html#mseevaluator)

      * [ParaphraseMiningEvaluator](evaluation.html#paraphraseminingevaluator)
      * [RerankingEvaluator](evaluation.html#rerankingevaluator)
      * [SentenceEvaluator](evaluation.html#sentenceevaluator)
      * [SequentialEvaluator](evaluation.html#sequentialevaluator)
      * [TranslationEvaluator](evaluation.html#translationevaluator)
      * [TripletEvaluator](evaluation.html#tripletevaluator)
    * Datasets
      * ParallelSentencesDataset
      * SentenceLabelDataset
      * DenoisingAutoEncoderDataset
      * NoDuplicatesDataLoader
    * [Models](models.html)
      * [Main Classes](models.html#main-classes)
      * [Further Classes](models.html#further-classes)
    * [quantization](quantization.html)
      * [`quantize_embeddings()`](quantization.html#sentence_transformers.quantization.quantize_embeddings)
      * [`semantic_search_faiss()`](quantization.html#sentence_transformers.quantization.semantic_search_faiss)
      * [`semantic_search_usearch()`](quantization.html#sentence_transformers.quantization.semantic_search_usearch)
  * [Cross Encoder](../cross_encoder/index.html)
    * [CrossEncoder](../cross_encoder/cross_encoder.html)
      * [CrossEncoder](../cross_encoder/cross_encoder.html#id1)

*

[`export_static_quantized_openvino_model()`](../util.html#sentence_transformers.backend.export_static_quantized_openvino_model)

* [Similarity Metrics](../util.html#module-sentence_transformers.util)
* [`cos_sim()`](../util.html#sentence_transformers.util.cos_sim)
* [`dot_score()`](../util.html#sentence_transformers.util.dot_score)
* [`euclidean_sim()`](../util.html#sentence_transformers.util.euclidean_sim)
* [`manhattan_sim()`](../util.html#sentence_transformers.util.manhattan_sim)
* [`pairwise_cos_sim()`](../util.html#sentence_transformers.util.pairwise_cos_sim)
* [`pairwise_dot_score()`](../util.html#sentence_transformers.util.pairwise_dot_score)
* [`pairwise_euclidean_sim()`](../util.html#sentence_transformers.util.pairwise_euclidean_sim)
* [`pairwise_manhattan_sim()`](../util.html#sentence_transformers.util.pairwise_manhattan_sim)

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

* [](../../../index.html)
* [Sentence Transformer](index.html)
* Datasets

* [Edit on GitHub](https://github.com/UKPLab/sentence-transformers/blob/master/docs/package_reference/sentence_transformer/datasets.md)

* * *

Datasetsïƒ•

Note

The `sentence_transformers.datasets` classes have been deprecated, and only exist for compatibility with the [deprecated training](../../sentence_transformer/training_overview.html#deprecated-training).

* Instead of `SentenceLabelDataset`, you can now use `BatchSamplers.GROUP_BY_LABEL` to use the [`GroupByLabelBatchSampler`](sampler.html#sentence_transformers.sampler.GroupByLabelBatchSampler "sentence_transformers.sampler.GroupByLabelBatchSampler").

* Instead of `NoDuplicatesDataLoader`, you can now use the `BatchSamplers.NO_DUPLICATES` to use the [`NoDuplicatesBatchSampler`](sampler.html#sentence_transformers.sampler.NoDuplicatesBatchSampler "sentence_transformers.sampler.NoDuplicatesBatchSampler").

`sentence_transformers.datasets` contains classes to organize your training input examples.

ParallelSentencesDatasetï*ƒ*•

`ParallelSentencesDataset` is used for multilingual training. For details, see [multilingual training](../../../examples/training/multilingual/README.html).

_class _sentence_transformers.datasets.ParallelSentencesDataset(_student_model:

[SentenceTransformer](SentenceTransformer.html#sentence_transformers.SentenceTransformer

"sentence_transformers.SentenceTransformer")_, _teacher_model :
[SentenceTransformer](SentenceTransformer.html#sentence_transformers.SentenceTransformer
"sentence_transformers.SentenceTransformer")_, _batch_size : int = 8_,
_use_embedding_cache : bool =
True_)[[source]](https://github.com/UKPLab/sentence-
transformers/blob/master/sentence_transformers\\datasets\\ParallelSentencesDataset.py#L25-L204
)ïƒ•

This dataset reader can be used to read-in parallel sentences, i.e., it reads
in a file with tab-seperated sentences with the same sentence in different
languages. For example, the file can look like this (EN DE ES): hello world
hallo welt hola mundo second sentence zweiter satz segunda oraciÃ³n

The sentence in the first column will be mapped to a sentence embedding using
the given the embedder. For example, embedder is a mono-lingual sentence
embedding method for English. The sentences in the other languages will also
be mapped to this English sentence embedding.

When getting a sample from the dataset, we get one sentence with the according
sentence embedding for this sentence.

teacher_model can be any class that implement an encode function. The encode
function gets a list of sentences and returns a list of sentence embeddings

Parallel sentences dataset reader to train student model given a teacher model

Parameters:

* **student_model** ([_SentenceTransformer_](SentenceTransformer.html#sentence_transformers.SentenceTransformer "sentence_transformers.SentenceTransformer")) â€“ The student sentence embedding model that should be trained.

* **teacher_model** ([_SentenceTransformer_](SentenceTransformer.html#sentence_transformers.SentenceTransformer "sentence_transformers.SentenceTransformer")) â€“ The teacher model that provides the sentence embeddings for the first column in the dataset file.

* **batch_size** (_int_ _,__optional_) â€“ The batch size for training. Defaults to 8.

* **use_embedding_cache** (_bool_ _,__optional_) â€“ Whether to use an embedding cache. Defaults to True.

SentenceLabelDatasetï*ƒ*•

`SentenceLabelDataset` can be used if you have labeled sentences and want to
train with triplet loss.

_class _sentence_transformers.datasets.SentenceLabelDataset(_examples :
list[[InputExample](../cross_encoder/cross_encoder.html#sentence_transformers.readers.InputExam

ple

"sentence_transformers.readers.InputExample.InputExample")]_,

_samples_per_label : int = 2_, _with_replacement : bool =

False_)[[source]](https://github.com/UKPLab/sentence-

transformers/blob/master/sentence_transformers\\datasets\\SentenceLabelDataset.py#L23-L111)ïƒ•

This dataset can be used for some specific Triplet Losses like

BATCH_HARD_TRIPLET_LOSS which requires multiple examples with the same label

in a batch.

It draws n consecutive, random and unique samples from one label at a time.

This is repeated for each label.

Labels with fewer than n unique samples are ignored. This also applied to

drawing without replacement, once less than n samples remain for a label, it

is skipped.

This _DOES NOT_ check if there are more labels than the batch is large or if

the batch size is divisible by the samples drawn per label.

Creates a LabelSampler for a SentenceLabelDataset.

Parameters:

* **examples** (_List_ _[_[_InputExample_](../cross_encoder/cross_encoder.html#sentence_transformers.readers.InputExample "sentence_transformers.readers.InputExample") _]_) â€“ A list of InputExamples.

* **samples_per_label** (_int_ _,__optional_) â€“ The number of consecutive, random, and unique samples drawn per label. The batch size should be a multiple of samples_per_label. Defaults to 2.

* **with_replacement** (_bool_ _,__optional_) â€“ If True, each sample is drawn at most once (depending on the total number of samples per label). If False, one sample can be drawn in multiple draws, but not multiple times in the same drawing. Defaults to False.

DenoisingAutoEncoderDatasetïƒ•

`DenoisingAutoEncoderDataset` is used for unsupervised training with the TSDAE method.

_class _sentence_transformers.datasets.DenoisingAutoEncoderDataset(_sentences: list[str], noise_fn=<function DenoisingAutoEncoderDataset.<lambda>>_)[[source]](https://github.com/UKPLab/sentence-transformers/blob/master/sentence_transformers\\datasets\\DenoisingAutoEncoderDataset.py#L21-L62)ïƒ•

The DenoisingAutoEncoderDataset returns InputExamples in the format: texts=[noise_fn(sentence), sentence] It is used in combination with the

DenoisingAutoEncoderLoss: Here, a decoder tries to re-construct the sentence without noise.

Parameters:

* **sentences** â€“ A list of sentences

* **noise_fn** â€“ A noise function: Given a string, it returns a string with noise, e.g. deleted words

NoDuplicatesDataLoaderïƒ•

`NoDuplicatesDataLoader`can be used together with MultipleNegativeRankingLoss to ensure that no duplicates are within the same batch.

_class _sentence_transformers.datasets.NoDuplicatesDataLoader(_train_examples_ , _batch_size_)[[source]](https://github.com/UKPLab/sentence-transformers/blob/master/sentence_transformers\\datasets\\NoDuplicatesDataLoader.py#L17-L56)ïƒ•

A special data loader to be used with MultipleNegativesRankingLoss. The data loader ensures that there are no duplicate sentences within the same batch

[Previous](evaluation.html "Evaluation") [Next](models.html "Models")

* * *

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).