```
[![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)

Sequence [Input Length](../../examples/applications/computing-embeddings/README.html#id1) [Multi-Process Multi-GPU / Encoding](../../examples/applications/computing-embeddings/README.html#multi-process-multi-g pu-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 Asymmetric Semantic VS. Search](../../examples/applications/semantic-search/README.html#symmetric-vs-asymmetric-se mantic-search) [Manual Implementation](../../examples/applications/semantic-search/README.html#manual-implementati on) [Optimized Implementation](../../examples/applications/semantic-search/README.html#optimized-implement ation) [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-neighb or) & [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-encode r) \* [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-retri eval) [Pre-trained Cross-Encoders (Re-Ranker)](../../examples/applications/retrieve\_rerank/README.html#pre-trained-cross-encoder s-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\_tr ansformers.util.paraphrase mining)

[Translated

Sentence

Mining](//examples/applications/parallel-sentence-mining/READM	E.html)	
*	[Margin	Based
Mining](//examples/applications/parallel-sentence-mining/READM	E.html#margin-ba	sed-mining)
* [Examples](//examples/applications/parallel-sentence-mining	ı/README.html#e	xamples)
* [Image Search](//examples/applications/image-search/READN	ИЕ.html)	
* [Installation](//examples/applications/image-search/README	.html#installation)	)
* [Usage](///examples/applications/image-search/README.htm	nl#usage)	
* [Examples](//examples/applications/image-search/README.	.html#examples)	
* [Embedding Quantization](//examples/applications/embedding	g-quantization/RE/	ADME.html)
	*	[Binary
Quantization](//examples/applications/embedding-quantization/RE	ADME.html#binar	y-quantizati
on)		
*	[Scalar	(int8)
Quantization](//examples/applications/embedding-quantization/RE	ADME.html#scala	ar-int8-quant
ization)		
	*	[Additional
extensions](//examples/applications/embedding-quantization/REA	DME.html#additio	nal-extensio
ns)		
* [Demo](//examples/applications/embedding-quantization/RE	ADME.html#demo	)
	* [Try	it
yourself](//examples/applications/embedding-quantization/READM	IE.html#try-it-your	self)
* [Speeding up Inference](//sentence_transformer/usage/efficience	y.html)	
* [PyTorch](//sentence_transformer/usage/efficiency.html#pytorc	ch)	
* [ONNX](//sentence_transformer/usage/efficiency.html#onnx)		
* [OpenVINO](//sentence_transformer/usage/efficiency.html#ope	envino)	
* [Benchmarks](//sentence_transformer/usage/efficiency.html#be	enchmarks)	

\* [Creating Custom Models](../../sentence\_transformer/usage/custom\_models.html)

- [Structure of Sentence Transformer Models](../../sentence\_transformer/usage/custom\_models.html#structure-of-sentence-transformer-m odels) [Sentence Transformer **Transformers** Model from а Model](../../sentence\_transformer/usage/custom\_models.html#sentence-transformer-model-from-a-t ransformers-model) \* [Pretrained Models](../../sentence\_transformer/pretrained\_models.html) \* [Original Models](../../sentence\_transformer/pretrained\_models.html#original-models) Search **[Semantic** Models](../../sentence\_transformer/pretrained\_models.html#semantic-search-models) \* [Multi-QA Models](../../sentence\_transformer/pretrained\_models.html#multi-ga-models) [MSMARCO Passage Models](../../sentence\_transformer/pretrained\_models.html#msmarco-passage-models) \* [Multilingual Models](../../sentence\_transformer/pretrained\_models.html#multilingual-models) [Semantic Similarity Models](../../sentence\_transformer/pretrained\_models.html#semantic-similarity-models) \* [Bitext Mining](../../sentence\_transformer/pretrained\_models.html#bitext-mining) \* [Image & Text-Models](../../sentence\_transformer/pretrained\_models.html#image-text-models) \* [INSTRUCTOR models](../../sentence transformer/pretrained models.html#instructor-models) [Scientific Similarity
- Models](../../sentence\_transformer/pretrained\_models.html#scientific-similarity-models)
  - \* [Training Overview](../../sentence\_transformer/training\_overview.html)
    - \* [Why Finetune?](../../sentence\_transformer/training\_overview.html#why-finetune)
    - \* [Training Components](../../sentence\_transformer/training\_overview.html#training-components)
    - \* [Dataset](../../sentence\_transformer/training\_overview.html#dataset)
      - \* [Dataset Format](../../sentence transformer/training overview.html#dataset-format)
  - \* [Loss Function](../../sentence\_transformer/training\_overview.html#loss-function)

\* [Training Arguments](../../sentence\_transformer/training\_overview.html#training-arguments) \* [Evaluator](../../sentence\_transformer/training\_overview.html#evaluator) \* [Trainer](../../sentence transformer/training overview.html#trainer) \* [Callbacks](../../sentence transformer/training overview.html#callbacks) \* [Multi-Dataset Training](../../sentence\_transformer/training\_overview.html#multi-dataset-training) \* [Deprecated Training](../../sentence\_transformer/training\_overview.html#deprecated-training) **Embedding** [Best Base Models](../../sentence transformer/training overview.html#best-base-embedding-models) \* [Dataset Overview](../../sentence transformer/dataset overview.html) [Datasets Hugging Face on the Hub](../../sentence\_transformer/dataset\_overview.html#datasets-on-the-hugging-face-hub) \* [Pre-existing Datasets](../../sentence\_transformer/dataset\_overview.html#pre-existing-datasets) \* [Loss Overview](../../sentence\_transformer/loss\_overview.html) \* [Loss modifiers](../../sentence transformer/loss overview.html#loss-modifiers) \* [Distillation](../../sentence\_transformer/loss\_overview.html#distillation) [Commonly used Loss Functions](../../sentence\_transformer/loss\_overview.html#commonly-used-loss-functions) \* [Custom Loss Functions](../../sentence\_transformer/loss\_overview.html#custom-loss-functions) \* [Training Examples](../../sentence transformer/training/examples.html)

- \* [Semantic Textual Similarity](../../examples/training/sts/README.html)
  - \* [Training data](../../examples/training/sts/README.html#training-data)
  - \* [Loss Function](../../examples/training/sts/README.html#loss-function)
- \* [Natural Language Inference](../../examples/training/nli/README.html)
  - \* [Data](../../examples/training/nli/README.html#data)
  - \* [SoftmaxLoss](../../examples/training/nli/README.html#softmaxloss)

- \* [Paraphrase Data](../../examples/training/paraphrases/README.html)
- \* [Pre-Trained Models](../../examples/training/paraphrases/README.html#pre-trained-models)
- \* [Quora Duplicate Questions](../../examples/training/quora\_duplicate\_questions/README.html)
  - \* [Training](../../examples/training/quora\_duplicate\_questions/README.html#training)

[MultipleNegativesRankingLoss](../../examples/training/quora\_duplicate\_questions/README.html# multiplenegativesrankingloss)

\* [Pretrained

Models](../../examples/training/quora\_duplicate\_questions/README.html#pretrained-models)

- \* [MS MARCO](../../examples/training/ms\_marco/README.html)
  - \* [Bi-Encoder](../../examples/training/ms\_marco/README.html#bi-encoder)
- \* [Matryoshka Embeddings](../../../examples/training/matryoshka/README.html)
  - \* [Use Cases](../../../examples/training/matryoshka/README.html#use-cases)
  - \* [Results](../../examples/training/matryoshka/README.html#results)
  - \* [Training](../../examples/training/matryoshka/README.html#training)
  - \* [Inference](../../examples/training/matryoshka/README.html#inference)
  - \* [Code Examples](../../examples/training/matryoshka/README.html#code-examples)
- \* [Adaptive Layers](../../examples/training/adaptive\_layer/README.html)
  - \* [Use Cases](../../../examples/training/adaptive layer/README.html#use-cases)
  - \* [Results](../../examples/training/adaptive\_layer/README.html#results)
  - \* [Training](../../examples/training/adaptive\_layer/README.html#training)
  - \* [Inference](../../../examples/training/adaptive\_layer/README.html#inference)
  - \* [Code Examples](../../examples/training/adaptive\_layer/README.html#code-examples)
- \* [Multilingual Models](../../examples/training/multilingual/README.html)

\* [Extend your own

models](../../examples/training/multilingual/README.html#extend-your-own-models)

\* [Training](../../examples/training/multilingual/README.html#training) \* [Datasets](../../examples/training/multilingual/README.html#datasets) [Sources for **Training** Data](../../examples/training/multilingual/README.html#sources-for-training-data) \* [Evaluation](../../examples/training/multilingual/README.html#evaluation) [Available Pre-trained Models](../../examples/training/multilingual/README.html#available-pre-trained-models) \* [Usage](../../examples/training/multilingual/README.html#usage) \* [Performance](../../examples/training/multilingual/README.html#performance) \* [Citation](../../examples/training/multilingual/README.html#citation) \* [Model Distillation](../../examples/training/distillation/README.html) [Knowledge Distillation](../../examples/training/distillation/README.html#knowledge-distillation) Performance [Speed Trade-Off](../../examples/training/distillation/README.html#speed-performance-trade-off) [Dimensionality Reduction](../../examples/training/distillation/README.html#dimensionality-reduction) \* [Quantization](../../examples/training/distillation/README.html#quantization) \* [Augmented SBERT](../../examples/training/data\_augmentation/README.html) \* [Motivation](../../examples/training/data\_augmentation/README.html#motivation) [Extend to vour own datasets](../../.examples/training/data\_augmentation/README.html#extend-to-your-own-datasets) \* [Methodology](../../examples/training/data\_augmentation/README.html#methodology) [Scenario 1: Limited or small annotated datasets (few labeled sentence-pairs)](../../examples/training/data\_augmentation/README.html#scenario-1-limited-or-s mall-annotated-datasets-few-labeled-sentence-pairs) [Scenario 2: No annotated datasets (Only unlabeled sentence-pairs)](../../examples/training/data\_augmentation/README.html#scenario-2-no-annotate d-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 train with we Prompts?](../../examples/training/prompts/README.html#why-would-we-train-with-prompts) [How do train with we 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 Pretrained а 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 Unsupervised VS. Learning](../../examples/domain adaptation/README.html#domain-adaptation-vs-unsupervised-le arning) [Adaptive Pre-Training](../../examples/domain\_adaptation/README.html#adaptive-pre-training) [GPL: Generative Pseudo-Labeling](../../examples/domain adaptation/README.html#gpl-generative-pseudo-labelin g) \* [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-encode r)

- \* [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-encoder s-re-ranker)

- \* [Pretrained Models](../../cross\_encoder/pretrained\_models.html)
  - \* [MS MARCO](../../cross\_encoder/pretrained\_models.html#ms-marco)
  - \* [SQuAD (QNLI)](../../cross\_encoder/pretrained\_models.html#squad-qnli)
  - \* [STSbenchmark](../../cross\_encoder/pretrained\_models.html#stsbenchmark)
    - \* [Quora Duplicate

Questions](../../cross\_encoder/pretrained\_models.html#quora-duplicate-questions)

- \* [NLI](../../cross\_encoder/pretrained\_models.html#nli)
- \* [Community Models](../../cross\_encoder/pretrained\_models.html#community-models)
- \* [Training Overview](../../cross\_encoder/training\_overview.html)
- \* [Training Examples](../../cross\_encoder/training/examples.html)
  - \* [MS MARCO](../../examples/training/ms\_marco/cross\_encoder\_README.html)

[Cross-Encoder](../../examples/training/ms\_marco/cross\_encoder\_README.html#cross-encoder)

\* [Cross-Encoder Knowledge

Distillation](../../examples/training/ms\_marco/cross\_encoder\_README.html#cross-encoder-knowledge-distillation)

Package Reference

- \* [Sentence Transformer](../sentence\_transformer/index.html)
  - \* [SentenceTransformer](../sentence\_transformer/SentenceTransformer.html)
    - \* [SentenceTransformer](../sentence\_transformer/SentenceTransformer.html#id1)

[SentenceTransformerModelCardData](../sentence\_transformer/SentenceTransformer.html#sentencetransformermodelcarddata)

- \* [SimilarityFunction](../sentence\_transformer/SentenceTransformer.html#similarityfunction)
- \* [Trainer](../sentence transformer/trainer.html)

[SentenceTransformerTrainer](../sentence\_transformer/trainer.html#sentencetransformertrainer)

\* [Training Arguments](../sentence\_transformer/training\_args.html)

[SentenceTransformerTrainingArguments](../sentence\_transformer/training\_args.html#sentencetran sformertrainingarguments)

- \* [Losses](../sentence\_transformer/losses.html)
  - \* [BatchAllTripletLoss](../sentence\_transformer/losses.html#batchalltripletloss)

[BatchHardSoftMarginTripletLoss](../sentence\_transformer/losses.html#batchhardsoftmargintripletloss)

- \* [BatchHardTripletLoss](../sentence\_transformer/losses.html#batchhardtripletloss)
- \* [BatchSemiHardTripletLoss](../sentence\_transformer/losses.html#batchsemihardtripletloss)
- \* [ContrastiveLoss](../sentence\_transformer/losses.html#contrastiveloss)
- \* [OnlineContrastiveLoss](../sentence\_transformer/losses.html#onlinecontrastiveloss)
- \* [ContrastiveTensionLoss](../sentence\_transformer/losses.html#contrastivetensionloss)

[ContrastiveTensionLossInBatchNegatives](../sentence\_transformer/losses.html#contrastivetensionl

\*

\*

ossinbatchnegatives)

- \* [CoSENTLoss](../sentence\_transformer/losses.html#cosentloss)
- \* [AnglELoss](../sentence\_transformer/losses.html#angleloss)
- \* [CosineSimilarityLoss](../sentence\_transformer/losses.html#cosinesimilarityloss)
- \* [DenoisingAutoEncoderLoss](../sentence\_transformer/losses.html#denoisingautoencoderloss)
- \* [GISTEmbedLoss](../sentence\_transformer/losses.html#gistembedloss)
- \* [CachedGISTEmbedLoss](../sentence\_transformer/losses.html#cachedgistembedloss)
- \* [MSELoss](../sentence\_transformer/losses.html#mseloss)
- \* [MarginMSELoss](../sentence transformer/losses.html#marginmseloss)
- \* [MatryoshkaLoss](../sentence\_transformer/losses.html#matryoshkaloss)
- \* [Matryoshka2dLoss](../sentence\_transformer/losses.html#matryoshka2dloss)
- \* [AdaptiveLayerLoss](../sentence\_transformer/losses.html#adaptivelayerloss)
- \* [MegaBatchMarginLoss](../sentence\_transformer/losses.html#megabatchmarginloss)

 $[Multiple Negatives Ranking Loss] (.../sentence\_transformer/losses.html \# multiple negatives ranking loss) \\$ 

[CachedMultipleNegativesRankingLoss](../sentence\_transformer/losses.html#cachedmultiplenegativesrankingloss)

[MultipleNegativesSymmetricRankingLoss](../sentence\_transformer/losses.html#multiplenegativessymmetricrankingloss)

[CachedMultipleNegativesSymmetricRankingLoss](../sentence\_transformer/losses.html#cachedmultiplenegativessymmetricrankingloss)

- \* [SoftmaxLoss](../sentence\_transformer/losses.html#softmaxloss)
- \* [TripletLoss](../sentence transformer/losses.html#tripletloss)
- \* [Samplers](../sentence\_transformer/sampler.html)

\*

\* [BatchSamplers](../sentence\_transformer/sampler.html#batchsamplers) [MultiDatasetBatchSamplers](../sentence transformer/sampler.html#multidatasetbatchsamplers) \* [Evaluation](../sentence transformer/evaluation.html) [BinaryClassificationEvaluator](../sentence\_transformer/evaluation.html#binaryclassificationevaluato [EmbeddingSimilarityEvaluator](../sentence transformer/evaluation.html#embeddingsimilarityevaluat or) [InformationRetrievalEvaluator](../sentence\_transformer/evaluation.html#informationretrievalevaluat or) \* [NanoBEIREvaluator](../sentence transformer/evaluation.html#nanobeirevaluator) \* [MSEEvaluator](../sentence\_transformer/evaluation.html#mseevaluator) [ParaphraseMiningEvaluator](../sentence\_transformer/evaluation.html#paraphraseminingevaluator) \* [RerankingEvaluator](../sentence\_transformer/evaluation.html#rerankingevaluator) \* [SentenceEvaluator](../sentence transformer/evaluation.html#sentenceevaluator) \* [SequentialEvaluator](../sentence transformer/evaluation.html#sequentialevaluator) \* [TranslationEvaluator](../sentence\_transformer/evaluation.html#translationevaluator) \* [TripletEvaluator](../sentence\_transformer/evaluation.html#tripletevaluator) \* [Datasets](../sentence\_transformer/datasets.html) \* [ParallelSentencesDataset](../sentence\_transformer/datasets.html#parallelsentencesdataset)

r)

[DenoisingAutoEncoderDataset](../sentence\_transformer/datasets.html#denoisingautoencoderdatas

\* [SentenceLabelDataset](../sentence\_transformer/datasets.html#sentencelabeldataset)

- \* [NoDuplicatesDataLoader](../sentence\_transformer/datasets.html#noduplicatesdataloader)
- \* [Models](../sentence\_transformer/models.html)
  - \* [Main Classes](../sentence\_transformer/models.html#main-classes)
  - \* [Further Classes](../sentence\_transformer/models.html#further-classes)
- \* [quantization](../sentence\_transformer/quantization.html)

[`quantize\_embeddings()`](../sentence\_transformer/quantization.html#sentence\_transformers.quantization.quantize embeddings)

[`semantic\_search\_faiss()`](../sentence\_transformer/quantization.html#sentence\_transformers.quantization.semantic\_search\_faiss)

[`semantic\_search\_usearch()`](../sentence\_transformer/quantization.html#sentence\_transformers.q uantization.semantic\_search\_usearch)

- \* [Cross Encoder](index.html)
  - \* [CrossEncoder](cross\_encoder.html)
    - \* [CrossEncoder](cross\_encoder.html#id1)
    - \* [Training Inputs](cross encoder.html#training-inputs)
  - \* Evaluation
    - \* CEBinaryAccuracyEvaluator
    - \* CEBinaryClassificationEvaluator
    - \* CECorrelationEvaluator
    - \* CEF1Evaluator
    - \* CESoftmaxAccuracyEvaluator
    - \* CERerankingEvaluator
- \* [util](../util.html)

- \* [Helper Functions](../util.html#module-sentence\_transformers.util)
  - \* [`community\_detection()`](../util.html#sentence\_transformers.util.community\_detection)
  - \* [`http\_get()`](../util.html#sentence\_transformers.util.http\_get)
  - \* [`is\_training\_available()`](../util.html#sentence\_transformers.util.is\_training\_available)
  - \* [`mine\_hard\_negatives()`](../util.html#sentence\_transformers.util.mine\_hard\_negatives)
  - \* [`normalize\_embeddings()`](../util.html#sentence\_transformers.util.normalize\_embeddings)
  - \* [`paraphrase\_mining()`](../util.html#sentence\_transformers.util.paraphrase\_mining)
  - \* [`semantic\_search()`](../util.html#sentence\_transformers.util.semantic\_search)
  - \* [`truncate\_embeddings()`](../util.html#sentence\_transformers.util.truncate\_embeddings)
- \* [Model Optimization](../util.html#module-sentence\_transformers.backend)

[`export\_dynamic\_quantized\_onnx\_model()`](../util.html#sentence\_transformers.backend.export\_dynamic\_quantized\_onnx\_model)

[`export\_optimized\_onnx\_model()`](../util.html#sentence\_transformers.backend.export\_optimized\_onnx\_model)

[`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)

,

```
__[Sentence Transformers](../../index.html)
 * [](../../index.html)
 * [Cross Encoder](index.html)
 * Evaluation
                                                       Edit
                                                                                                on
GitHub](https://github.com/UKPLab/sentence-transformers/blob/master/docs/package_reference/cro
ss_encoder/evaluation.md)
# Evaluationïf•
CrossEncoder have their own evaluation classes, that are in
`sentence_transformers.cross_encoder.evaluation`.
## CEBinaryAccuracyEvaluatorif•
_class
_sentence_transformers.cross_encoder.evaluation.CEBinaryAccuracyEvaluator(_sentence_pairs
: list[list[str]]_, _labels : list[int]_, _name : str = "_, _threshold :
float = 0.5_, _write_csv : bool =
True_)[[source]](https://github.com/UKPLab/sentence-
transformers/blob/master/sentence_transformers\\cross_encoder\\evaluation\\CEBinaryAccuracyEv
aluator.py#L14-L80)ïf•
```

\* [`pairwise\_manhattan\_sim()`](../util.html#sentence\_transformers.util.pairwise\_manhattan\_sim)

This evaluator can be used with the CrossEncoder class.

It is designed for CrossEncoders with 1 outputs. It measure the accuracy of the predict class vs. the gold labels. It uses a fixed threshold to determine the label (0 vs 1).

See CEBinaryClassificationEvaluator for an evaluator that determines automatically the optimal threshold.

## CEBinaryClassificationEvaluatorif•

```
class
```

\_sentence\_transformers.cross\_encoder.evaluation.CEBinaryClassificationEvaluator(\_sentence\_pair s

```
: list[list[str]]\_, \_labels : list[int]\_, \_name : str = "\_, \\
```

\_show\_progress\_bar : bool = False\_, \_write\_csv : bool =

True\_)[[source]](https://github.com/UKPLab/sentence-

transformers/blob/master/sentence\_transformers\\cross\_encoder\\evaluation\\CEBinaryClassification{
nEvaluator.py#L16-L104)ïf•

This evaluator can be used with the CrossEncoder class. Given sentence pairs and binary labels (0 and 1), it compute the average precision and the best

```
possible f1 score
```

```
## CECorrelationEvaluatorïf•
```

```
_class 
_sentence_transformers.cross_encoder.evaluation.CECorrelationEvaluator(_sentence_pairs : list[list[str]]_, _scores : list[float]_, _name : str = "_, _write_csv : bool = True_)[[source]](https://github.com/UKPLab/sentence-transformers/blob/master/sentence_transformers\\cross_encoder\\evaluation\\CECorrelationEvaluat or.py#L14-L67)\ddot{r}f^{\bullet}
```

This evaluator can be used with the CrossEncoder class. Given sentence pairs and continuous scores, it compute the pearson & spearman correlation between the predicted score for the sentence pair and the gold score.

```
## CEF1Evaluatorïf•
```

class

```
_sentence_transformers.cross_encoder.evaluation.CEF1Evaluator(_sentence_pairs : list[list[str]]_, _labels : list[int]_, _*_ , _batch_size : int = 32_, _show_progress_bar : bool = False_, _name : str = "_, _write_csv : bool = True_)[[source]](https://github.com/UKPLab/sentence-transformers/blob/master/sentence_transformers\\cross_encoder\\evaluation\\CEF1Evaluator.py#L1 6-L140)\ddot{i}f^{\bullet}
```

CrossEncoder F1 score based evaluator for binary and multiclass tasks.

The task type (binary or multiclass) is determined from the labels array. For binary tasks the returned metric is binary F1 score. For the multiclass tasks the returned metric is macro F1 score.

## Parameters:

\* \*\*sentence\_pairs\*\* (\_List\_ \_[\_\_List\_ \_[\_\_str\_ \_]\_\_]\_) – A list of sentence pairs, where each pair is a list of two strings.

- \* \*\*labels\*\* (\_List\_ \_[\_\_int\_ \_]\_) â€" A list of integer labels corresponding to each sentence pair.
- \* \*\*batch\_size\*\* (\_int\_ \_,\_\_optional\_) â€" Batch size for prediction. Defaults to 32.
- \* \*\*show\_progress\_bar\*\* (\_bool\_ \_,\_\_optional\_) â€" Show tqdm progress bar.
- \* \*\*name\*\* (\_str\_ \_,\_\_optional\_) â€" An optional name for the CSV file with stored results. Defaults to an empty string.
- \* \*\*write\_csv\*\* (\_bool\_ \_,\_\_optional\_) â€" Flag to determine if the data should be saved to a CSV file. Defaults to True.

## CESoftmaxAccuracyEvaluatorif•

```
class
```

\_sentence\_transformers.cross\_encoder.evaluation.CESoftmaxAccuracyEvaluator(\_sentence\_pairs

: list[list[str]]\_, \_labels : list[int]\_, \_name : str = "\_, \_write\_csv : bool

= True\_)[[source]](https://github.com/UKPLab/sentence-

 $transformers/blob/master/sentence\_transformers\cross\_encoder\evaluation\CESoftmaxAccuracyEvaluator.py\#L14-L70)\"{i}f \bullet$ 

This evaluator can be used with the CrossEncoder class.

It is designed for CrossEncoders with 2 or more outputs. It measure the accuracy of the predict class vs. the gold labels.

## CERerankingEvaluatorïf•

\_class \_sentence\_transformers.cross\_encoder.evaluation.CERerankingEvaluator(\_samples\_ , \_at\_k : int = 10\_, \_name : str = "\_, \_write\_csv : bool = True\_, \_mrr\_at\_k : int | None = None\_)[[source]](https://github.com/UKPLab/sentence-transformers/blob/master/sentence\_transformers\\cross\_encoder\\evaluation\\CERerankingEvaluator.py#L13-L109)if•

This class evaluates a CrossEncoder model for the task of re-ranking.

Given a query and a list of documents, it computes the score [query, doc\_i]

for all possible documents and sorts them in decreasing order. Then, [MRR@10](/cdn-cgi/l/emailprotection#bff2eded999c8c8884999c8a8d84999c8b87848e8f) and [NDCG@10](/cdncgi/l/email-protection#df919b9c98f9fcece8e4f9fceaede4f9fcebe7e4eeef) are computed to measure the quality of the ranking. Parameters: \*\*samples\*\* (\_List\_ \_[\_\_Dict\_ \_,\_\_str\_ \_,\_\_Union\_ \_[\_\_str\_ \_,\_\_List\_ \_[\_\_str\_ \_]\_\_]\_) â€" Must be a list and each element is of the form: {â€~query': â€~〙, â€~positive': [], â€~negative': []}. Query is the search query, positive is a list of positive (relevant) documents, negative is a list of negative (irrelevant) documents. [ Previous](cross\_encoder.html "CrossEncoder") [Next ](../util.html "util") (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).