

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

## Getting Started

- \* [Installation](../../../../docs/installation.html)

- \* [Install with pip](../../../../docs/installation.html#install-with-pip)

- \* [Install with Conda](../../../../docs/installation.html#install-with-conda)

- \* [Install from Source](../../../../docs/installation.html#install-from-source)

- \* [Editable Install](../../../../docs/installation.html#editable-install)

- \* [Install PyTorch with CUDA support](../../../../docs/installation.html#install-pytorch-with-cuda-support)

- \* [Quickstart](../../../../docs/quickstart.html)

- \* [Sentence Transformer](../../../../docs/quickstart.html#sentence-transformer)

- \* [Cross Encoder](../../../../docs/quickstart.html#cross-encoder)

- \* [Next Steps](../../../../docs/quickstart.html#next-steps)

## Sentence Transformer

- \* [Usage](../../../../docs/sentence\_transformer/usage/usage.html)

- \* Computing Embeddings

- \* Initializing a Sentence Transformer Model

- \* Calculating Embeddings

- \* Prompt Templates

- \* Input Sequence Length

- \* 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](../semantic-search/README.html)

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

\*

[Symmetric

vs.

Asymmetric

Semantic

Search](../semantic-search/README.html#symmetric-vs-asymmetric-semantic-search)

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

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

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

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

\*

[Approximate

Nearest

Neighbor](../semantic-search/README.html#approximate-nearest-neighbor)

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

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

\* [Retrieve & Re-Rank](../retrieve\_rerank/README.html)

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

\* [Retrieval: Bi-Encoder](../retrieve\_rerank/README.html#retrieval-bi-encoder)

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

\* [Example Scripts](../retrieve\_rerank/README.html#example-scripts)

\*

[Pre-trained

Bi-Encoders

(Retrieval)](../retrieve\_rerank/README.html#pre-trained-bi-encoders-retrieval)

\*

[Pre-trained

Cross-Encoders

(Re-Ranker)](../retrieve\_rerank/README.html#pre-trained-cross-encoders-re-ranker)

\* [Clustering](../clustering/README.html)

\* [k-Means](../clustering/README.html#k-means)

\* [Agglomerative Clustering](../clustering/README.html#agglomerative-clustering)

\* [Fast Clustering](../clustering/README.html#fast-clustering)

\* [Topic Modeling](../clustering/README.html#topic-modeling)

\* [Paraphrase Mining](../paraphrase-mining/README.html)

\*

[`paraphrase\_mining()`](../paraphrase-mining/README.html#sentence\_transformers.util.paraphrase\_mining)

\* [Translated Sentence Mining](../parallel-sentence-mining/README.html)

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

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

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

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

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

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

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

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

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

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

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

\* [Try it yourself](../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-transfo

mer-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/README.html)

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

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

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

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

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

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

\* [Example Scripts](../../training/hpo/README.html#example-scripts)

\* [Distributed Training](../../docs/sentence\_transformer/training/distributed.html)

\* [Comparison](../../docs/sentence\_transformer/training/distributed.html#comparison)

\* [FSDP](../../docs/sentence\_transformer/training/distributed.html#fsdp)

## Cross Encoder

\* [Usage](../../docs/cross\_encoder/usage/usage.html)

\* [Retrieve & Re-Rank](../retrieve\_rerank/README.html)

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

\* [Retrieval: Bi-Encoder](../retrieve\_rerank/README.html#retrieval-bi-encoder)

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

\* [Example Scripts](../retrieve\_rerank/README.html#example-scripts)

\* [Pre-trained Bi-Encoders (Retrieval)](../retrieve\_rerank/README.html#pre-trained-bi-encoders-retrieval)

\* [Pre-trained Cross-Encoders (Re-Ranker)](../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#batchhardsoftmargintripletloss)

\*

[BatchHardTripletLoss](../../docs/package\_reference/sentence\_transformer/losses.html#batchhardtripletloss)

\*

[BatchSemiHardTripletLoss](../../docs/package\_reference/sentence\_transformer/losses.html#batchsemihardtripletloss)

\*

[ContrastiveLoss](../../docs/package\_reference/sentence\_transformer/losses.html#contrastiveloss)

\*

[OnlineContrastiveLoss](../../docs/package\_reference/sentence\_transformer/losses.html#onlinecontrastiveloss)

\*

[ContrastiveTensionLoss](../../docs/package\_reference/sentence\_transformer/losses.html#contrastivetensionloss)

\*

[ContrastiveTensionLossInBatchNegatives](../../docs/package\_reference/sentence\_transformer/losses.html#contrastivetensionlossinbatchnegatives)

\* [CoSENTLoss](../../docs/package\_reference/sentence\_transformer/losses.html#cosentloss)

\* [AngleLoss](../../docs/package\_reference/sentence\_transformer/losses.html#angleloss)

\*

[CosineSimilarityLoss](../../docs/package\_reference/sentence\_transformer/losses.html#cosinesimilarityloss)

\*

[DenoisingAutoEncoderLoss](../../docs/package\_reference/sentence\_transformer/losses.html#denoisingautoencoderloss)

\*

[GISTEmbedLoss](../../docs/package\_reference/sentence\_transformer/losses.html#gistembedloss)

\*

[CachedGISTEmbedLoss](../../docs/package\_reference/sentence\_transformer/losses.html#cachedgistembedloss)

\* [MSELoss](../../docs/package\_reference/sentence\_transformer/losses.html#mseloss)

\*

[MarginMSELoss](../../docs/package\_reference/sentence\_transformer/losses.html#marginmseloss)

\*

[MatryoshkaLoss](../../docs/package\_reference/sentence\_transformer/losses.html#matryoshkaloss)

\*

[Matryoshka2dLoss](../../docs/package\_reference/sentence\_transformer/losses.html#matryoshka2dloss)

\*

[AdaptiveLayerLoss](../../docs/package\_reference/sentence\_transformer/losses.html#adaptivelayerloss)

\*

[MegaBatchMarginLoss](../../docs/package\_reference/sentence\_transformer/losses.html#megabatchmarginloss)

\*

[MultipleNegativesRankingLoss](../../docs/package\_reference/sentence\_transformer/losses.html#multiplenegativesrankingloss)

\*

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

\*

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

\*

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

\*

\* [SoftmaxLoss](../../docs/package\_reference/sentence\_transformer/losses.html#softmaxloss)

\* [TripletLoss](../../docs/package\_reference/sentence\_transformer/losses.html#tripletloss)

\* [Samplers](../../docs/package\_reference/sentence\_transformer/sampler.html)

\*

[BatchSamplers](../../../../docs/package\_reference/sentence\_transformer/sampler.html#batchsamplers)

\*

[MultiDatasetBatchSamplers](../../../../docs/package\_reference/sentence\_transformer/sampler.html#multidatasetbatchsamplers)

\* [Evaluation](../../../../docs/package\_reference/sentence\_transformer/evaluation.html)

\*

[BinaryClassificationEvaluator](../../../../docs/package\_reference/sentence\_transformer/evaluation.html#binaryclassificationevaluator)

\*

[EmbeddingSimilarityEvaluator](../../../../docs/package\_reference/sentence\_transformer/evaluation.html#embeddingsimilarityevaluator)

\*

[InformationRetrievalEvaluator](../../../../docs/package\_reference/sentence\_transformer/evaluation.html#informationretrievalevaluator)

\*

[NanoBEIREvaluator](../../../../docs/package\_reference/sentence\_transformer/evaluation.html#nanobeirevaluator)

\*

[MSEEvaluator](../../../../docs/package\_reference/sentence\_transformer/evaluation.html#mseevaluator)

\*

[ParaphraseMiningEvaluator](../../../../docs/package\_reference/sentence\_transformer/evaluation.html#paraphraseminingevaluator)

\*

[RerankingEvaluator](../../../../docs/package\_reference/sentence\_transformer/evaluation.html#rerankingevaluator)

\*

[SentenceEvaluator](../../docs/package\_reference/sentence\_transformer/evaluation.html#sentenceevaluator)

\*

[SequentialEvaluator](../../docs/package\_reference/sentence\_transformer/evaluation.html#sequentialevaluator)

\*

[TranslationEvaluator](../../docs/package\_reference/sentence\_transformer/evaluation.html#translationevaluator)

\*

[TripletEvaluator](../../docs/package\_reference/sentence\_transformer/evaluation.html#tripletevaluator)

\* [Datasets](../../docs/package\_reference/sentence\_transformer/datasets.html)

\*

[ParallelSentencesDataset](../../docs/package\_reference/sentence\_transformer/datasets.html#parallelsentencesdataset)

\*

[SentenceLabelDataset](../../docs/package\_reference/sentence\_transformer/datasets.html#sentencelabeldataset)

\*

[DenoisingAutoEncoderDataset](../../docs/package\_reference/sentence\_transformer/datasets.html#denoisingautoencoderdataset)

\*

[NoDuplicatesDataLoader](../../docs/package\_reference/sentence\_transformer/datasets.html#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)



nevaluator)

\* [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\_sim)

\*

[`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)

\* [Usage](../../docs/sentence\_transformer/usage/usage.html)

\* Computing Embeddings

\*

[

Edit

on

GitHub](https://github.com/UKPLab/sentence-transformers/blob/master/examples/applications/computing-embeddings/README.rst)

\* \* \*

# Computing Embeddings¶

Once you have [installed](installation.md) Sentence Transformers, you can easily use Sentence Transformer models:

1.

```
[`SentenceTransformer`](../../docs/package_reference/sentence_transformer/SentenceTransformer.html#sentence_transformers.SentenceTransformer  
"sentence_transformers.SentenceTransformer")
```

2.

```
[`SentenceTransformer.encode`](../../docs/package_reference/sentence_transformer/SentenceTransformer.html#sentence_transformers.SentenceTransformer.encode  
"sentence_transformers.SentenceTransformer.encode")
```

3.

```
[`SentenceTransformer.similarity`](../../docs/package_reference/sentence_transformer/SentenceTransformer.html#sentence_transformers.SentenceTransformer.similarity  
"sentence_transformers.SentenceTransformer.similarity")
```

```
from sentence_transformers import SentenceTransformer
```

```
# 1. Load a pretrained Sentence Transformer model
```

```
model = SentenceTransformer("all-MiniLM-L6-v2")
```

```
# The sentences to encode
```

```
sentences = [
```

```
    "The weather is lovely today.",
```

```
"It's so sunny outside!",  
"He drove to the stadium.",  
]
```

```
# 2. Calculate embeddings by calling model.encode()  
embeddings = model.encode(sentences)  
print(embeddings.shape)  
# [3, 384]  
  
# 3. Calculate the embedding similarities  
similarities = model.similarity(embeddings, embeddings)  
print(similarities)  
# tensor([[1.0000, 0.6660, 0.1046],  
#         [0.6660, 1.0000, 0.1411],  
#         [0.1046, 0.1411, 1.0000]])
```

## Note

Even though we talk about sentence embeddings, you can use Sentence Transformers for shorter phrases as well as for longer texts with multiple sentences. See [Input Sequence Length](#) for notes on embeddings for longer texts.

## ## Initializing a Sentence Transformer Model

The first step is to load a pretrained Sentence Transformer model. You can use any of the models from the [\[Pretrained](#)

Models](../docs/sentence\_transformer/pretrained\_models.html) or a local model.

See also

[`SentenceTransformer`](../../docs/package\_reference/sentence\_transformer/SentenceTransformer.html#sentence\_transformers.SentenceTransformer

"sentence\_transformers.SentenceTransformer") for information on parameters.

```
from sentence_transformers import SentenceTransformer
```

```
model = SentenceTransformer("all-mpnet-base-v2")
```

```
# Alternatively, you can pass a path to a local model directory:
```

```
model = SentenceTransformer("output/models/mpnet-base-finetuned-all-nli")
```

The model will automatically be placed on the most performant available device, e.g. `cuda` or `mps` if available. You can also specify the device explicitly:

```
model = SentenceTransformer("all-mpnet-base-v2", device="cuda")
```

### ## Calculating Embeddings

The method to calculate embeddings is `SentenceTransformer.encode`.

## ## Prompt Templates

Some models require using specific text `_prompts_` to achieve optimal performance. For example, with `[intfloat/multilingual-e5-large](https://huggingface.co/intfloat/multilingual-e5-large)` you should prefix all queries with ``"query: "`` and all passages with ``"passage: "``. Another example is `[BAAI/bge-large-en-v1.5](https://huggingface.co/BAAI/bge-large-en-v1.5)`, which performs best for retrieval when the input texts are prefixed with ``"Represent this sentence for searching relevant passages: "``.

Sentence Transformer models can be initialized with ``prompts`` and ``default_prompt_name`` parameters:

\* ``prompts`` is an optional argument that accepts a dictionary of prompts with prompt names to prompt texts. The prompt will be prepended to the input text during inference. For example:

```
model = SentenceTransformer(  
    "intfloat/multilingual-e5-large",  
    prompts={  
        "classification": "Classify the following text: ",  
        "retrieval": "Retrieve semantically similar text: ",  
        "clustering": "Identify the topic or theme based on the text: ",  
    },  
)  
# or
```

```

model.prompts = {
    "classification": "Classify the following text: ",
    "retrieval": "Retrieve semantically similar text: ",
    "clustering": "Identify the topic or theme based on the text: ",
}

```

\* `default_prompt_name` is an optional argument that determines the default prompt to be used. It has to correspond with a prompt name from `prompts`. If `None`, then no prompt is used by default. For example:

```

model = SentenceTransformer(
    "intfloat/multilingual-e5-large",
    prompts={
        "classification": "Classify the following text: ",
        "retrieval": "Retrieve semantically similar text: ",
        "clustering": "Identify the topic or theme based on the text: ",
    },
    default_prompt_name="retrieval",
)
# or
model.default_prompt_name="retrieval"

```

Both of these parameters can also be specified in the

`config_sentence_transformers.json` file of a saved model. That way, you

won't have to specify these options manually when loading. When you save a



Sentence Transformer model, these options will be automatically saved as well.

During inference, prompts can be applied in a few different ways. All of these scenarios result in identical texts being embedded:

1. Explicitly using the ``prompt`` option in ``SentenceTransformer.encode``:

```
embeddings = model.encode("How to bake a strawberry cake", prompt="Retrieve semantically similar text: ")
```

2. Explicitly using the ``prompt_name`` option in ``SentenceTransformer.encode`` by relying on the prompts loaded from a) initialization or b) the model config:

```
embeddings = model.encode("How to bake a strawberry cake", prompt_name="retrieval")
```

3. If ``prompt`` nor ``prompt_name`` are specified in ``SentenceTransformer.encode``, then the prompt specified by ``default_prompt_name`` will be applied. If it is ``None``, then no prompt will be applied:

```
embeddings = model.encode("How to bake a strawberry cake")
```

## ## Input Sequence Length

For transformer models like BERT, RoBERTa, DistilBERT etc., the runtime and memory requirement grows quadratic with the input length. This limits

transformers to inputs of certain lengths. A common value for BERT-based models are 512 tokens, which corresponds to about 300-400 words (for English).

Each model has a maximum sequence length under `model.max_seq_length``, which is the maximal number of tokens that can be processed. Longer texts will be truncated to the first `model.max_seq_length`` tokens:

```
from sentence_transformers import SentenceTransformer
```

```
model = SentenceTransformer("all-MiniLM-L6-v2")
```

```
print("Max Sequence Length:", model.max_seq_length)
```

```
# => Max Sequence Length: 256
```

```
# Change the length to 200
```

```
model.max_seq_length = 200
```

```
print("Max Sequence Length:", model.max_seq_length)
```

```
# => Max Sequence Length: 200
```

## Note

You cannot increase the length higher than what is maximally supported by the respective transformer model. Also note that if a model was trained on short texts, the representations for long texts might not be that good.

## ## Multi-Process / Multi-GPU Encoding

You can encode input texts with more than one GPU (or with multiple processes on a CPU machine). For an example, see:

[computing\_embeddings\_multi\_gpu.py](https://github.com/UKPLab/sentence-transformers/blob/master/examples/applications/computing-embeddings/computing\_embeddings\_multi\_gpu.py).

The relevant method is

[start\_multi\_process\_pool()](../docs/package\_reference/sentence\_transformer/SentenceTransformer.html#sentence\_transformers.SentenceTransformer.start\_multi\_process\_pool "sentence\_transformers.SentenceTransformer.start\_multi\_process\_pool"), which starts multiple processes that are used for encoding.

[ Previous](../docs/sentence\_transformer/usage/usage.html "Usage") [Next](../docs/sentence\_transformer/usage/semantic\_textual\_similarity.html "Semantic Textual Similarity")

\* \* \*

(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).

