![Logo](_static/logo.png)
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](examples/applications/computing-embeddings/README.html)
* [Initializing a Sentence Transforme
Model](examples/applications/computing-embeddings/README.html#initializing-a-sentence-transfo
rmer-model)
* [Calculatin
Embeddings](examples/applications/computing-embeddings/README.html#calculating-embeddings
s)
* [Promp
Templates](examples/applications/computing-embeddings/README.html#prompt-templates)

* [Input Sequence Length](examples/applications/compu	ting-embeddi	ngs/READM	E.html#id1)
* [N	/lulti-Process	/	Multi-GPU
Encoding](examples/applications/computing-embeddings/RE	EADME.html#	multi-proces	s-multi-gpu-en
coding)			
*	[Ser	nantic	Textual
Similarity](docs/sentence_transformer/usage/semantic_textu	ıal_similarity.l	ntml)	
		*	[Similarity
Calculation](docs/sentence_transformer/usage/semantic_tex	ktual_similarit	y.html#simila	rity-calculatio
n)			
* [Semantic Search](examples/applications/semantic-sear	ch/README.	html)	
* [Background](examples/applications/semantic-search/F	README.htm	l#backgroun	d)
* [Symmetric	vs. A	Asymmetric	Semantic
Search](examples/applications/semantic-search/README.h	tml#symmetri	c-vs-asymm	etric-semantic
-search)			
		*	[Manual
Implementation](examples/applications/semantic-search/RE	ADME.html#r	manual-imple	ementation)
		*	[Optimized
Implementation](examples/applications/semantic-search/RE	ADME.html#c	ptimized-imp	olementation)
		*	[Speed
Optimization](examples/applications/semantic-search/READ	ME.html#spe	ed-optimizat	ion)
* [Elasticsearch](examples/applications/semantic-search	/README.htr	ml#elasticsea	arch)
*	[App	proximate	Nearest
Neighbor](examples/applications/semantic-search/README	.html#approx	imate-neares	st-neighbor)
* [Retrieve & Re-Rank](examples/applications/semantic-	search/READ	ME.html#ret	rieve-re-rank)
* [Examples](examples/applications/semantic-search/RE	ADME.html#	examples)	
* [Retrieve & Re-Rank](examples/applications/retrieve_ren	rank/READMI	∃.html)	
*	[Retrieve	&	Re-Rank

Pipeline](examples/applications/retrieve_rerank/README.html#retrieve	-re-rank-pipel	line)
	*	[Retrieval:
Bi-Encoder](examples/applications/retrieve_rerank/README.html#retrieve_	eval-bi-encod	er)
	*	[Re-Ranker:
Cross-Encoder](examples/applications/retrieve_rerank/README.html#	re-ranker-cros	ss-encoder)
* [Example Scripts](examples/applications/retrieve_rerank/README	E.html#examp	le-scripts)
* [Pre	-trained	Bi-Encoders
(Retrieval)](examples/applications/retrieve_rerank/README.html#pre-t	rained-bi-enco	oders-retrieval)
* [Pre-tra	ined (Cross-Encoders
(Re-Ranker)](examples/applications/retrieve_rerank/README.html#pre	e-trained-cross	s-encoders-re-r
anker)		
* [Clustering](examples/applications/clustering/README.html)		
* [k-Means](examples/applications/clustering/README.html#k-mea	ns)	
	*	[Agglomerative
Clustering](examples/applications/clustering/README.html#agglomera	tive-clustering	j)
* [Fast Clustering](examples/applications/clustering/README.html#	fast-clusterin	g)
* [Topic Modeling](examples/applications/clustering/README.html#	topic-modelir	ng)
* [Paraphrase Mining](examples/applications/paraphrase-mining/REA	ADME.html)	
		*
[`paraphrase_mining()`](examples/applications/paraphrase-mining/REA	.DME.html#se	entence_transfo
rmers.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)

* [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-quantizatio n) [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](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 Transformer of Sentence Models](docs/sentence transformer/usage/custom models.html#structure-of-sentence-transformermodels) [Sentence Transformer Model from Transformers а 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

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

Models](docs/sentence_transformer/pretrained_models)	els.html#sem	nantic-search-model	s)
* [Multi-QA Models](docs/sentence_transformer/p	retrained_m	odels.html#multi-qa	-models)
	*	[MSMARCO	Passage
Models](docs/sentence_transformer/pretrained_models	els.html#msn	narco-passage-moc	lels)
* [Multilingual Models](docs/sentence_transformer/	pretrained_r	models.html#multilin	gual-models)
	*	[Semantic	Similarity
Models](docs/sentence_transformer/pretrained_models	els.html#sem	nantic-similarity-mod	lels)
* [Bitext Mining](docs/sentence_transformer/pretra	ained_mode	ls.html#bitext-minin	g)
* [Image & Text-Models](docs/sentence_transform	er/pretrained	l_models.html#imag	je-text-models)
* [INSTRUCTOR models](docs/sentence_transform	ner/pretraine	ed_models.html#inst	ructor-models)
	*	[Scientific	Similarity
Models](docs/sentence_transformer/pretrained_models	els.html#scie	entific-similarity-mod	els)
* [Training Overview](docs/sentence_transformer/tra	nining_overvi	iew.html)	
* [Why Finetune?](docs/sentence_transformer/train	ning_overvie	w.html#why-finetune	e)
		*	[Training
Components](docs/sentence_transformer/training_ov	erview.html#	training-component	ts)
* [Dataset](docs/sentence_transformer/training_over	erview.html#	dataset)	
* [Dataset Format](docs/sentence_transformer/tra	ining_overvi	iew.html#dataset-fo	rmat)
* [Loss Function](docs/sentence_transformer/traini	ng_overview	v.html#loss-function))
* [Training Arguments](docs/sentence_transformer	/training_ove	erview.html#training	-arguments)
* [Evaluator](docs/sentence_transformer/training_c	verview.htm	l#evaluator)	
* [Trainer](docs/sentence_transformer/training_ove	rview.html#t	rainer)	
* [Callbacks](docs/sentence_transformer/training_	_overview.ht	ml#callbacks)	
		*	[Multi-Dataset
Training](docs/sentence_transformer/training_overvie	w.html#mult	:i-dataset-training)	
* [Deprecated Training](docs/sentence_transforme	r/training_ov	erview.html#depred	ated-training)
*	[Best	Base	Embedding

Models](docs/sentence_transformer/tra	aining_overviev	v.html#best	-base-emb	edding-models))
* [Dataset Overview](docs/sentence_	transformer/dat	taset_overv	riew.html)		
*	[Datasets	on	the	Hugging	Face
Hub](docs/sentence_transformer/datas	set_overview.ht	tml#dataset	s-on-the-h	ugging-face-hub	o)
			*	[Pr	e-existing
Datasets](docs/sentence_transformer/	dataset_overvi	ew.html#pre	e-existing-c	datasets)	
* [Loss Overview](docs/sentence_train	nsformer/loss_c	overview.hti	ml)		
* [Loss modifiers](docs/sentence_tra	ansformer/loss_	_overview.h	tml#loss-m	nodifiers)	
* [Distillation](docs/sentence_transfo	ormer/loss_ove	rview.html#	distillation)		
	*	[Commo	only	used	Loss
Functions](docs/sentence_transformer	:/loss_overview	.html#comn	nonly-used	I-loss-functions)	l
		*	[C	ustom	Loss
Functions](docs/sentence_transformer	/loss_overview	.html#custo	m-loss-fun	ctions)	
* [Training Examples](docs/sentence	_transformer/tra	aining/exam	ples.html)		
* [Semantic Textual Similarity](exam	ples/training/st	s/README	.html)		
* [Training data](examples/training/	/sts/README.h	ntml#trainin	g-data)		
* [Loss Function](examples/training	g/sts/README.	html#loss-f	unction)		
* [Natural Language Inference](exar	nples/training/n	ıli/README	i.html)		
* [Data](examples/training/nli/REAI	DME.html#data	ı)			
* [SoftmaxLoss](examples/training/	/nli/README.ht	tml#softmax	kloss)		
					*
[MultipleNegativesRankingLoss](exam	ples/training/nli	i/README.l	html#multip	olenegativesran	kingloss)
* [Paraphrase Data](examples/traini	ng/paraphrases	s/README.	.html)		
* [Pre-Trained Models](examples/ti	raining/paraphr	ases/READ	ME.html#p	ore-trained-mod	els)

 $\hbox{* [Quora Duplicate Questions](examples/training/quora_duplicate_questions/README.html)}\\$

* [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) [Speed Performance 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 your own datasets](examples/training/data_augmentation/README.html#extend-to-your-own-datasets) * [Methodology](examples/training/data_augmentation/README.html#methodology) [Scenario 1: Limited small annotated datasets labeled or (few sentence-pairs)](examples/training/data_augmentation/README.html#scenario-1-limited-or-small-a nnotated-datasets-few-labeled-sentence-pairs) [Scenario 2: No annotated datasets (Only unlabeled sentence-pairs)](examples/training/data_augmentation/README.html#scenario-2-no-annotated-dat asets-only-unlabeled-sentence-pairs) * [Training](examples/training/data_augmentation/README.html#training)

* [Citation](examples/training/data_augmentation/README.html#citation)

* [What are Prompts?](examples/training/prompts/README.html#what-are-prompts)

[Why

would

train

we

with

* [Training with Prompts](examples/training/prompts/README.html)

Prompts?](examples/training/prompts/RE	EADME.h	tml#why-\	vould-we-tra	in-with-prom	pts)	
	*	[How	do	we	train	with
Prompts?](examples/training/prompts/RE	EADME.h	tml#how-o	do-we-train-v	vith-prompts))	
* [Training with PEFT Adapters](exam	ples/train	ing/peft/R	EADME.htm	1)		
* [Compatibility Methods](examples/t	raining/pe	eft/READN	/IE.html#com	npatibility-me	thods)	
* [Adding a New Adapter](examples/	training/p	eft/READI	ME.html#ado	ling-a-new-a	dapter)	
		*	[Loading	а	Pref	trained
Adapter](examples/training/peft/READM	E.html#lo	ading-a-p	retrained-ada	apter)		
* [Training Script](examples/training/	peft/REAI	OME.html	#training-scri	ipt)		
* [Unsupervised Learning](examples/u	ınsupervi	sed_learn	ing/README	E.html)		
* [TSDAE](examples/unsupervised_l	earning/F	README.	ntml#tsdae)			
* [SimCSE](examples/unsupervised_	_learning/	README	.html#simcse))		
* [CT](examples/unsupervised_learn	ing/READ	ME.html#	tct)			
		*	[CT	(In-Batch	Ne	egative
Sampling)](examples/unsupervised_lear	ning/REA	DME.htm	l#ct-in-batch	-negative-sa	mpling)	
		*	Masked	Languag	je	Model
(MLM)](examples/unsupervised_learning	J/READM	E.html#m	asked-langu	age-model-m	ılm)	
* [GenQ](examples/unsupervised_lea	arning/RE	ADME.ht	ml#genq)			
* [GPL](examples/unsupervised_lear	ning/REA	DME.htm	l#gpl)			
				*	[Perfor	mance
Comparison](examples/unsupervised_le	arning/RE	EADME.ht	ml#performa	nce-compar	ison)	
* [Domain Adaptation](examples/doma	ain_adapf	ation/REA	ADME.html)			
*	[Dor	nain	Adaptation	vs.	Unsupe	ervised
Learning](examples/domain_adaptation/	README	.html#dor	nain-adaptat	ion-vs-unsup	ervised-le	earnin
g)						
* [Adaptive Pre-Training](examples/d	omain_a	daptation/	README.htr	ml#adaptive-	pre-trainir	ng)
			*	[GPL:	Gen	erative

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](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](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](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](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](docs/package_reference/sentence_transformer/index.html)

[SentenceTransformer](docs/package_reference/sentence_transformer/SentenceTransformer.html)

[SentenceTransformer](docs/package_reference/sentence_transformer/SentenceTransformer.html#i d1)

[SentenceTransformerModelCardData](docs/package_reference/sentence_transformer/SentenceTransformer.html#sentencetransformermodelcarddata)

[SimilarityFunction](docs/package_reference/sentence_transformer/SentenceTransformer.html#simil

arityfunction) * [Trainer](docs/package_reference/sentence_transformer/trainer.html) [SentenceTransformerTrainer](docs/package_reference/sentence_transformer/trainer.html#sentence etransformertrainer) * [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#batc hhardsoftmargintripletloss) [BatchHardTripletLoss](docs/package_reference/sentence_transformer/losses.html#batchhardtripletl oss) [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)

 $[Contrastive Tension Loss] (docs/package_reference/sentence_transformer/losses.html \# contrastive tension Loss) (docs/package_reference/sentence_transformer/losses.html \# contrastive tension Losses) (docs/package_reference/sentence_transformer/losses) (docs/package_reference/sentenc$

nsionloss)

*

[ContrastiveTensionLossInBatchNegatives](docs/package_reference/sentence_transformer/losses.h tml#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)

l#denoisina

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

* [GISTEmbedLoss](docs/package_reference/sentence_transformer/losses.html#gistembedloss)

*

[CachedGISTEmbedLoss](docs/package_reference/sentence_transformer/losses.html#cachedgiste mbedloss)

- * [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)

ŀ

[Matryoshka2dLoss](docs/package_reference/sentence_transformer/losses.html#matryoshka2dloss)

k

[AdaptiveLayerLoss](docs/package_reference/sentence_transformer/losses.html#adaptivelayerloss)

*

[MegaBatchMarginLoss](docs/package_reference/sentence_transformer/losses.html#megabatchma

rginloss)

[MultipleNegativesRankingLoss](docs/package_reference/sentence_transformer/losses.html#multipl

enegativesrankingloss)

[CachedMultipleNegativesRankingLoss](docs/package_reference/sentence_transformer/losses.html

#cachedmultiplenegativesrankingloss)

[MultipleNegativesSymmetricRankingLoss](docs/package_reference/sentence_transformer/losses.h

tml#multiplenegativessymmetricrankingloss)

[CachedMultipleNegativesSymmetricRankingLoss](docs/package_reference/sentence_transformer/l

osses.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#multidat

asetbatchsamplers)

* [Evaluation](docs/package_reference/sentence_transformer/evaluation.html)

[BinaryClassificationEvaluator](docs/package_reference/sentence_transformer/evaluation.html#bina

ryclassificationevaluator)

[EmbeddingSimilarityEvaluator](docs/package_reference/sentence_transformer/evaluation.html#em

beddingsimilarityevaluator)

*

•

[InformationRetrievalEvaluator](docs/package_reference/sentence_transformer/evaluation.html#info rmationretrievalevaluator) [NanoBEIREvaluator](docs/package_reference/sentence_transformer/evaluation.html#nanobeireval uator) [MSEEvaluator](docs/package_reference/sentence_transformer/evaluation.html#mseevaluator) [ParaphraseMiningEvaluator](docs/package_reference/sentence_transformer/evaluation.html#parap hraseminingevaluator) [RerankingEvaluator](docs/package_reference/sentence_transformer/evaluation.html#rerankingeval uator) [SentenceEvaluator](docs/package_reference/sentence_transformer/evaluation.html#sentenceevalu ator) [SequentialEvaluator](docs/package_reference/sentence_transformer/evaluation.html#sequentialev aluator)

[TranslationEvaluator](docs/package_reference/sentence_transformer/evaluation.html#translationevaluator)

 $[Triplet Evaluator] (docs/package_reference/sentence_transformer/evaluation.html \# triplet evaluator) \\$

^{* [}Datasets](docs/package_reference/sentence_transformer/datasets.html)

[ParallelSentencesDataset](docs/package_reference/sentence_transformer/datasets.html#parallelse ntencesdataset)

*

[SentenceLabelDataset](docs/package_reference/sentence_transformer/datasets.html#sentencelab eldataset)

ŧ

[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)

t

[`semantic_search_faiss()`](docs/package_reference/sentence_transformer/quantization.html#sente nce_transformers.quantization.semantic_search_faiss)

*

[`semantic_search_usearch()`](docs/package_reference/sentence_transformer/quantization.html#se ntence_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#cebinary classificationevaluator)

[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#cesoftmax accuracyevaluator)

[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_a vailable)

*

[`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)

k

[`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)

ŕ

[`export_dynamic_quantized_onnx_model()`](docs/package_reference/util.html#sentence_transform ers.backend.export_dynamic_quantized_onnx_model)

•

[`export_optimized_onnx_model()`](docs/package_reference/util.html#sentence_transformers.backe nd.export_optimized_onnx_model)

r

[`export_static_quantized_openvino_model()`](docs/package_reference/util.html#sentence_transfor mers.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
* * SentenceTransformers Documentation * [Edit on GitHub](https://github.com/UKPLab/sentence-transformers/blob/master/index.rst)
* * *

You can now use ONNX and OpenVINO backends for Sentence Transformer models.

Read [SentenceTransformer > Usage > Speeding up

Inference](docs/sentence_transformer/usage/efficiency.html) to learn more

about the new backends and what they can mean for your inference speed.

Tip

Sentence Transformers v3.4 just released, introducing documentation for training with PEFT. Read [SentenceTransformer > Training Examples > Training with PEFT Adapters](examples/training/peft/README.html) to learn more about how you can use train embedding models without finetuning all model parameters.

SentenceTransformers Documentation if •

Sentence Transformers (a.k.a. SBERT) is the go-to Python module for accessing, using, and training state-of-the-art text and image embedding models. It can be used to compute embeddings using Sentence Transformer models ([quickstart](docs/quickstart.html#sentence-transformer)) or to calculate similarity scores using Cross-Encoder models ([quickstart](docs/quickstart.html#cross-encoder)). This unlocks a wide range of applications, including [semantic search](examples/applications/semantic-search/README.html), [semantic textual similarity](docs/usage/semantic_textual_similarity.html), and [paraphrase mining](examples/applications/paraphrase-mining/README.html).

A wide selection of over [5,000 pre-trained Sentence Transformers models](https://huggingface.co/models?library=sentence-transformers) are available for immediate use on 🤗 Hugging Face, including many of the state-of-the-art models from the [Massive Text Embeddings Benchmark (MTEB) leaderboard](https://huggingface.co/spaces/mteb/leaderboard). Additionally, it is easy to [train or finetune your own models](docs/sentence_transformer/training_overview.html) using Sentence Transformers, enabling you to create custom models for your specific use cases.

Sentence Transformers was created by [UKPLab](http://www.ukp.tu-darmstadt.de/) and is being maintained by [🤗 Hugging Face](https://huggingface.co).

Don't hesitate to open an issue on the [Sentence Transformers repository](https://github.com/UKPLab/sentence-transformers) if something is broken or if you have further questions.

Usageïf•

See also

See the [Quickstart](docs/quickstart.html) for more quick information on how to use Sentence Transformers.

Using Sentence Transformer models is elementary:

Installation

```
You can install _sentence-transformers_ using pip:
  pip install -U sentence-transformers
We recommend **Python 3.9+** and **PyTorch 1.11.0+**. See
[installation](docs/installation.html) for further installation options.
  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. 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]])
```

What Next?ïf•

Consider reading one of the following sections to answer the related questions:

- * How to **use** Sentence Transformer models? [Sentence Transformers > Usage](docs/sentence_transformer/usage/usage.html)
- * What Sentence Transformer **models** can I use? [Sentence Transformers > Pretrained Models](docs/sentence_transformer/pretrained_models.html)
- * How do I make Sentence Transformer models **faster**? [Sentence Transformers > Usage > Speeding up Inference](docs/sentence_transformer/usage/efficiency.html)
- * How do I **train/finetune** a Sentence Transformer model? [Sentence Transformers > Training

 Overview](docs/sentence_transformer/training_overview.html)

```
Cross
                                               Encoder
                                                           models?
                                                                        [Cross
                                                                                  Encoder
             How
                     to
                           **use**
Usage](docs/cross_encoder/usage/usage.html)
        What Cross Encoder
                                  **models**
                                              can I use?
                                                              [Cross
                                                                       Encoder
                                                                                     Pretrained
Models](docs/cross_encoder/pretrained_models.html)
# Citingïf•
If you find this repository helpful, feel free to cite our publication
[Sentence-BERT: Sentence Embeddings using Siamese BERT-
Networks](https://arxiv.org/abs/1908.10084):
>
    @inproceedings{reimers-2019-sentence-bert,
>
     title = "Sentence-BERT: Sentence Embeddings using Siamese BERT-
> Networks",
     author = "Reimers, Nils and Gurevych, Iryna",
     booktitle = "Proceedings of the 2019 Conference on Empirical Methods
> in Natural Language Processing",
     month = "11",
>
     year = "2019",
>
     publisher = "Association for Computational Linguistics",
>
     url = "https://arxiv.org/abs/1908.10084",
>
>
   }
>
```

If you use one of the multilingual models, feel free to cite our publication

[Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation](https://arxiv.org/abs/2004.09813):

```
>
    @inproceedings{reimers-2020-multilingual-sentence-bert,
>
     title = "Making Monolingual Sentence Embeddings Multilingual using
>
> Knowledge Distillation",
     author = "Reimers, Nils and Gurevych, Iryna",
     booktitle = "Proceedings of the 2020 Conference on Empirical Methods
> in Natural Language Processing",
     month = "11",
>
     year = "2020",
>
     publisher = "Association for Computational Linguistics",
>
     url = "https://arxiv.org/abs/2004.09813",
>
    }
>
If you use the code for [data
augmentation](https://github.com/UKPLab/sentence-
transformers/tree/master/examples/training/data_augmentation), feel free to
cite our publication [Augmented SBERT: Data Augmentation Method for Improving
Bi-Encoders for Pairwise Sentence Scoring
```

> @inproceedings{thakur-2020-AugSBERT,

Tasks](https://arxiv.org/abs/2010.08240):

>

> title = "Augmented {SBERT}: Data Augmentation Method for Improving Bi-

- > Encoders for Pairwise Sentence Scoring Tasks",
- > author = "Thakur, Nandan and Reimers, Nils and Daxenberger, Johannes
- > and Gurevych, Iryna",
- > booktitle = "Proceedings of the 2021 Conference of the North American
- > Chapter of the Association for Computational Linguistics: Human Language
- > Technologies",
- > month = jun,
- > year = "2021",
- > address = "Online",
- > publisher = "Association for Computational Linguistics",
- > url = "https://www.aclweb.org/anthology/2021.naacl-main.28",
- > pages = "296--310",
- > }

>

[Next](docs/installation.html "Installation")

* * *

(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).