Transformers

English | 简体中文 | 繁體中文 | 한국어

State-of-the-art Machine Learning for JAX, PyTorch and TensorFlow

🤗 Transformers provides thousands of pretrained models to perform tasks on different modalities such as text, vision, and audio.

These models can be applied on:

- 📝 Text, for tasks like text classification, information extraction, question answering, summarization, translation, text generation, in over 100 languages.
- 🖼️ Images, for tasks like image classification, object detection, and segmentation.
- 🗣️ Audio, for tasks like speech recognition and audio classification.

Transformer models can also perform tasks on **several modalities combined**, such as table question answering, optical character recognition, information extraction from scanned documents, video classification, and visual question answering.

🤗 Transformers provides APIs to quickly download and use those pretrained models on a given text, fine-tune them on your own datasets and then share them with the community on our model hub. At the same time, each python module defining an architecture is fully standalone and can be modified to enable quick research experiments.

🤗 Transformers is backed by the three most popular deep learning libraries — Jax, PyTorch and TensorFlow — with a seamless integration between them. It's straightforward to train your models with one before loading them for inference with the other.

Online demos

You can test most of our models directly on their pages from the model hub. We also offer private model hosting, versioning, & an inference API for public and private models.

Here are a few examples:

In Natural Language Processing:

- Masked word completion with BERT
- Name Entity Recognition with Electra
- Text generation with GPT-2
- Natural Language Inference with RoBERTa
- Summarization with BART
- Question answering with DistilBERT
- Translation with T5

In Computer Vision:

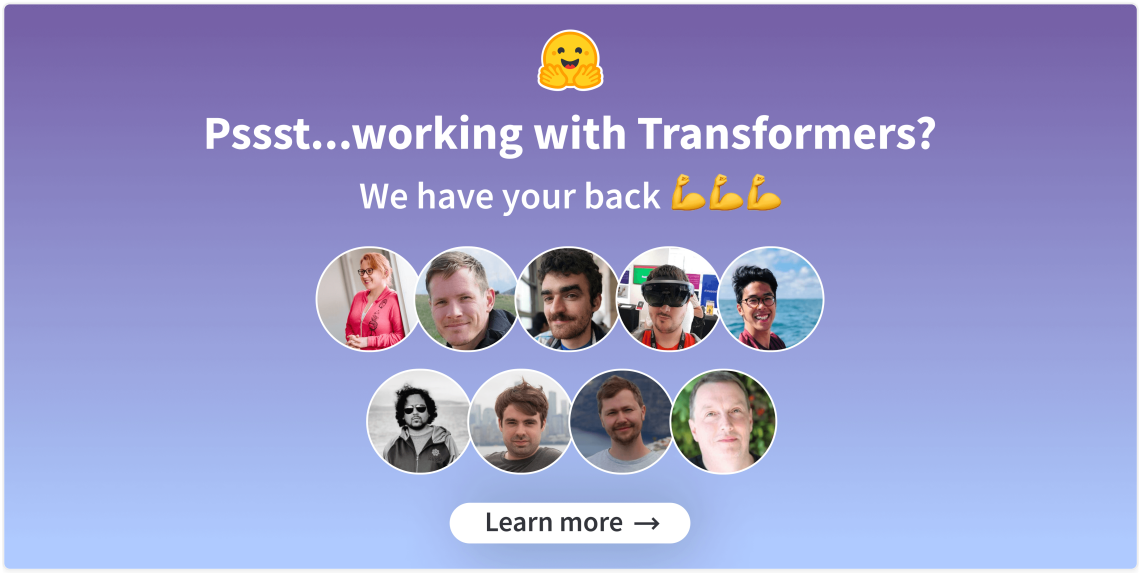
- Image classification with ViT
- Object Detection with DETR
- Image Segmentation with DETR

In Audio:

- Automatic Speech Recognition with Wav2Vec2
- Keyword Spotting with Wav2Vec2

Write With Transformer, built by the Hugging Face team, is the official demo of this repo's text generation capabilities.

If you are looking for custom support from the Hugging Face team



Quick tour

To immediately use a model on a given input (text, image, audio, ...), we provide the `pipeline` API. Pipelines group together a pretrained model with the preprocessing that was used during that model's training. Here is how to quickly use a pipeline to classify positive versus negative texts:

```
>>> from transformers import pipeline

# Allocate a pipeline for sentiment-analysis
>>> classifier = pipeline('sentiment-analysis')
>>> classifier('We are very happy to introduce pipeline to the transformers repository.')
[{'label': 'POSITIVE', 'score': 0.9996980428695679}]
```

The second line of code downloads and caches the pretrained model used by the pipeline, while the third evaluates it on the given text. Here the answer is "positive" with a confidence of 99.97%.

Many NLP tasks have a pre-trained `pipeline` ready to go. For example, we can easily extract question answers given context:

```
>>> from transformers import pipeline

# Allocate a pipeline for question-answering
>>> question_answerer = pipeline('question-answering')
>>> question_answerer({
...     'question': 'What is the name of the repository ?',
...     'context': 'Pipeline has been included in the huggingface/transformers
repository'
... })
{'score': 0.30970096588134766, 'start': 34, 'end': 58, 'answer':
'huggingface/transformers'}
```

In addition to the answer, the pretrained model used here returned its confidence score, along with the start position and end position of the answer in the tokenized sentence. You can learn more about the tasks supported by the `pipeline` API in this tutorial.

To download and use any of the pretrained models on your given task, all it takes is three lines of code. Here is the PyTorch version:

```
>>> from transformers import AutoTokenizer, AutoModel

>>> tokenizer = AutoTokenizer.from_pretrained("bert-base-uncased")
>>> model = AutoModel.from_pretrained("bert-base-uncased")

>>> inputs = tokenizer("Hello world!", return_tensors="pt")
>>> outputs = model(**inputs)
```

And here is the equivalent code for TensorFlow:

```
>>> from transformers import AutoTokenizer, TFAutoModel

>>> tokenizer = AutoTokenizer.from_pretrained("bert-base-uncased")
>>> model = TFAutoModel.from_pretrained("bert-base-uncased")

>>> inputs = tokenizer("Hello world!", return_tensors="tf")
>>> outputs = model(**inputs)
```

The tokenizer is responsible for all the preprocessing the pretrained model expects, and can be called directly on a single string (as in the above examples) or a list. It will output a dictionary that you can use in downstream code or simply directly pass to your model using the ** argument unpacking operator.

The model itself is a regular Pytorch `nn.Module` or a TensorFlow `tf.keras.Model` (depending on your backend) which you can use normally. This tutorial explains how to integrate such a model into a classic PyTorch or TensorFlow training loop, or how to use our `Trainer` API to quickly fine-tune on a new dataset.

Why should I use transformers?

1. Easy-to-use state-of-the-art models:
    - High performance on natural language understanding & generation, computer vision, and audio tasks.
    - Low barrier to entry for educators and practitioners.
    - Few user-facing abstractions with just three classes to learn.
    - A unified API for using all our pretrained models.
2. Lower compute costs, smaller carbon footprint:
    - Researchers can share trained models instead of always retraining.
    - Practitioners can reduce compute time and production costs.
    - Dozens of architectures with over 20,000 pretrained models, some in more than 100 languages.
3. Choose the right framework for every part of a model's lifetime:
    - Train state-of-the-art models in 3 lines of code.
    - Move a single model between TF2.0/PyTorch/JAX frameworks at will.
    - Seamlessly pick the right framework for training, evaluation and production.
4. Easily customize a model or an example to your needs:
    - We provide examples for each architecture to reproduce the results published by its original authors.
    - Model internals are exposed as consistently as possible.
    - Model files can be used independently of the library for quick experiments.

Why shouldn't I use transformers?

- This library is not a modular toolbox of building blocks for neural nets. The code in the model files is not refactored with additional abstractions on purpose, so that researchers can quickly iterate on each of the models without diving into additional abstractions/files.
- The training API is not intended to work on any model but is optimized to work with the models provided by the library. For generic machine learning loops, you should use another library.
- While we strive to present as many use cases as possible, the scripts in our examples folder are just that: examples. It is expected that they won't work out-of-the box on your specific problem and that you will be required to change a few lines of code to adapt them to your needs.

Installation

With pip

This repository is tested on Python 3.6+, Flax 0.3.2+, PyTorch 1.3.1+ and TensorFlow 2.3+.

You should install 🤗 Transformers in a virtual environment. If you're unfamiliar with Python virtual environments, check out the user guide.

First, create a virtual environment with the version of Python you're going to use and activate it.

Then, you will need to install at least one of Flax, PyTorch or TensorFlow. Please refer to TensorFlow installation page, PyTorch installation page and/or Flax and Jax installation pages regarding the specific install command for your platform.

When one of those backends has been installed, 🤗 Transformers can be installed using pip as follows:

```
pip install transformers
```

If you'd like to play with the examples or need the bleeding edge of the code and can't wait for a new release, you must install the library from source.

With conda

Since Transformers version v4.0.0, we now have a conda channel: `huggingface`.

🤗 Transformers can be installed using conda as follows:

```
conda install -c huggingface transformers
```

Follow the installation pages of Flax, PyTorch or TensorFlow to see how to install them with conda.

Model architectures

All the model checkpoints provided by 🤗 Transformers are seamlessly integrated from the huggingface.co model hub where they are uploaded directly by users and organizations.

Current number of checkpoints: 

🤗 Transformers currently provides the following architectures (see here for a high-level summary of each them):

1. **ALBERT** (from Google Research and the Toyota Technological Institute at Chicago) released with the paper ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, by Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut.
2. **BART** (from Facebook) released with the paper BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension by Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov and Luke Zettlemoyer.
3. **BARThez** (from École polytechnique) released with the paper BARThez: a Skilled Pretrained French Sequence-to-Sequence Model by Moussa Kamal Eddine, Antoine J.-P. Tixier, Michalis Vazirgiannis.
4. **BARTpho** (from VinAI Research) released with the paper BARTpho: Pre-trained Sequence-to-Sequence Models for Vietnamese by Nguyen Luong Tran, Duong Minh Le and Dat Quoc Nguyen.
5. **BEiT** (from Microsoft) released with the paper BEiT: BERT Pre-Training of Image Transformers by Hangbo Bao, Li Dong, Furu Wei.
6. **BERT** (from Google) released with the paper BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding by Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova.
7. **BERTweet** (from VinAI Research) released with the paper BERTweet: A pre-trained language model for English Tweets by Dat Quoc Nguyen, Thanh Vu and Anh Tuan Nguyen.
8. **BERT For Sequence Generation** (from Google) released with the paper Leveraging Pre-trained Checkpoints for Sequence Generation Tasks by Sascha Rothe, Shashi Narayan, Aliaksei Severyn.
9. **BigBird-RoBERTa** (from Google Research) released with the paper Big Bird: Transformers for Longer Sequences by Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, Amr Ahmed.
10. **BigBird-Pegasus** (from Google Research) released with the paper Big Bird: Transformers for Longer Sequences by Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, Amr Ahmed.
11. **Blenderbot** (from Facebook) released with the paper Recipes for building an open-domain chatbot by Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M. Smith, Y-Lan Boureau, Jason Weston.
12. **BlenderbotSmall** (from Facebook) released with the paper Recipes for building an open-domain chatbot by Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M. Smith, Y-Lan Boureau, Jason Weston.
13. **BORT** (from Alexa) released with the paper Optimal Subarchitecture Extraction For BERT by Adrian de Wynter and Daniel J. Perry.
14. **ByT5** (from Google Research) released with the paper ByT5: Towards a token-free future with pre-trained byte-to-byte models by Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, Colin Raffel.

15. **CamemBERT** (from Inria/Facebook/Sorbonne) released with the paper CamemBERT: a Tasty French Language Model by Louis Martin*, Benjamin Muller*, Pedro Javier Ortiz Suárez*, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah and Benoît Sagot.
16. **CANINE** (from Google Research) released with the paper CANINE: Pre-training an Efficient Tokenization-Free Encoder for Language Representation by Jonathan H. Clark, Dan Garrette, Iulia Turc, John Wieting.
17. **ConvNeXT** (from Facebook AI) released with the paper A ConvNet for the 2020s by Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, Saining Xie.
18. **CLIP** (from OpenAI) released with the paper Learning Transferable Visual Models From Natural Language Supervision by Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, Ilya Sutskever.
19. **ConvBERT** (from YituTech) released with the paper ConvBERT: Improving BERT with Span-based Dynamic Convolution by Zihang Jiang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, Shuicheng Yan.
20. **CPM** (from Tsinghua University) released with the paper CPM: A Large-scale Generative Chinese Pre-trained Language Model by Zhengyan Zhang, Xu Han, Hao Zhou, Pei Ke, Yuxian Gu, Deming Ye, Yujia Qin, Yusheng Su, Haozhe Ji, Jian Guan, Fanchao Qi, Xiaozhi Wang, Yanan Zheng, Guoyang Zeng, Huanqi Cao, Shengqi Chen, Daixuan Li, Zhenbo Sun, Zhiyuan Liu, Minlie Huang, Wentao Han, Jie Tang, Juanzi Li, Xiaoyan Zhu, Maosong Sun.
21. **CTRL** (from Salesforce) released with the paper CTRL: A Conditional Transformer Language Model for Controllable Generation by Nitish Shirish Keskar*, Bryan McCann*, Lav R. Varshney, Caiming Xiong and Richard Socher.
22. **Data2Vec** (from Facebook) released with the paper Data2Vec: A General Framework for Self-supervised Learning in Speech, Vision and Language by Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, Michael Auli.
23. **DeBERTa** (from Microsoft) released with the paper DeBERTa: Decoding-enhanced BERT with Disentangled Attention by Pengcheng He, Xiaodong Liu, Jianfeng Gao, Weizhu Chen.
24. **DeBERTa-v2** (from Microsoft) released with the paper DeBERTa: Decoding-enhanced BERT with Disentangled Attention by Pengcheng He, Xiaodong Liu, Jianfeng Gao, Weizhu Chen.
25. **Decision Transformer** (from Berkeley/Facebook/Google) released with the paper Decision Transformer: Reinforcement Learning via Sequence Modeling by Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, Igor Mordatch.
26. **DiT** (from Microsoft Research) released with the paper DiT: Self-supervised Pre-training for Document Image Transformer by Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, Furu Wei.
27. **DeiT** (from Facebook) released with the paper Training data-efficient image transformers & distillation through attention by Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, Hervé Jégou.
28. **DETR** (from Facebook) released with the paper End-to-End Object Detection with Transformers by Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, Sergey Zagoruyko.
29. **DialoGPT** (from Microsoft Research) released with the paper DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation by Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, Bill Dolan.
30. **DistilBERT** (from HuggingFace), released together with the paper DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter by Victor Sanh, Lysandre Debut and Thomas Wolf. The same method has been applied to compress GPT2 into DistilGPT2, RoBERTa into DistilRoBERTa, Multilingual BERT into DistilmBERT and a German version of DistilBERT.
31. **DPR** (from Facebook) released with the paper Dense Passage Retrieval for Open-Domain Question Answering by Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih.
32. **DPT** (from Intel Labs) released with the paper Vision Transformers for Dense Prediction by René Ranftl, Alexey Bochkovskiy, Vladlen Koltun.
33. **EncoderDecoder** (from Google Research) released with the paper Leveraging Pre-trained Checkpoints for Sequence Generation Tasks by Sascha Rothe, Shashi Narayan, Aliaksei Severyn.
34. **ELECTRA** (from Google Research/Stanford University) released with the paper ELECTRA: Pre-training text encoders as discriminators rather than generators by Kevin Clark, Minh-Thang Luong, Quoc V. Le, Christopher D. Manning.
35. **FlauBERT** (from CNRS) released with the paper FlauBERT: Unsupervised Language Model Pre-training for French by Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, Didier Schwab.

36. **FNet** (from Google Research) released with the paper FNet: Mixing Tokens with Fourier Transforms by James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, Santiago Ontanon.
37. **Funnel Transformer** (from CMU/Google Brain) released with the paper Funnel-Transformer: Filtering out Sequential Redundancy for Efficient Language Processing by Zihang Dai, Guokun Lai, Yiming Yang, Quoc V. Le.
38. **GLPN** (from KAIST) released with the paper Global-Local Path Networks for Monocular Depth Estimation with Vertical CutDepth by Doyeon Kim, Woonghyun Ga, Pyungwhan Ahn, Donggyu Joo, Sehwan Chun, Junmo Kim.
39. **GPT** (from OpenAI) released with the paper Improving Language Understanding by Generative Pre-Training by Alec Radford, Karthik Narasimhan, Tim Salimans and Ilya Sutskever.
40. **GPT-2** (from OpenAI) released with the paper Language Models are Unsupervised Multitask Learners by Alec Radford*, Jeffrey Wu*, Rewon Child, David Luan, Dario Amodei** and Ilya Sutskever**.
41. **GPT-J** (from EleutherAI) released in the repository kingoflolz/mesh-transformer-jax by Ben Wang and Aran Komatsuzaki.
42. **GPT Neo** (from EleutherAI) released in the repository EleutherAI/gpt-neo by Sid Black, Stella Biderman, Leo Gao, Phil Wang and Connor Leahy.
43. **Hubert** (from Facebook) released with the paper HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units by Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, Abdelrahman Mohamed.
44. **I-BERT** (from Berkeley) released with the paper I-BERT: Integer-only BERT Quantization by Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W. Mahoney, Kurt Keutzer.
45. **ImageGPT** (from OpenAI) released with the paper Generative Pretraining from Pixels by Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, Ilya Sutskever.
46. **LayoutLM** (from Microsoft Research Asia) released with the paper LayoutLM: Pre-training of Text and Layout for Document Image Understanding by Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou.
47. **LayoutLMv2** (from Microsoft Research Asia) released with the paper LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding by Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, Lidong Zhou.
48. **LayoutXLM** (from Microsoft Research Asia) released with the paper LayoutXLM: Multimodal Pre-training for Multilingual Visually-rich Document Understanding by Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Furu Wei.
49. **LED** (from AllenAI) released with the paper Longformer: The Long-Document Transformer by Iz Beltagy, Matthew E. Peters, Arman Cohan.
50. **Longformer** (from AllenAI) released with the paper Longformer: The Long-Document Transformer by Iz Beltagy, Matthew E. Peters, Arman Cohan.
51. **LUKE** (from Studio Ousia) released with the paper LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention by Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, Yuji Matsumoto.
52. **mLUKE** (from Studio Ousia) released with the paper mLUKE: The Power of Entity Representations in Multilingual Pretrained Language Models by Ryokan Ri, Ikuya Yamada, and Yoshimasa Tsuruoka.
53. **LXMERT** (from UNC Chapel Hill) released with the paper LXMERT: Learning Cross-Modality Encoder Representations from Transformers for Open-Domain Question Answering by Hao Tan and Mohit Bansal.
54. **M2M100** (from Facebook) released with the paper Beyond English-Centric Multilingual Machine Translation by Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, Armand Joulin.
55. **MarianMT** Machine translation models trained using OPUS data by Jörg Tiedemann. The Marian Framework is being developed by the Microsoft Translator Team.
56. **MaskFormer** (from Meta and UIUC) released with the paper Per-Pixel Classification is Not All You Need for Semantic Segmentation by Bowen Cheng, Alexander G. Schwing, Alexander Kirillov.
57. **MBart** (from Facebook) released with the paper Multilingual Denoising Pre-training for Neural Machine Translation by Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, Luke Zettlemoyer.
58. **MBart-50** (from Facebook) released with the paper Multilingual Translation with Extensible Multilingual Pretraining and Finetuning by Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, Angela Fan.

59. **Megatron-BERT** (from NVIDIA) released with the paper Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism by Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper and Bryan Catanzaro.
60. **Megatron-GPT2** (from NVIDIA) released with the paper Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism by Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper and Bryan Catanzaro.
61. **MPNet** (from Microsoft Research) released with the paper MPNet: Masked and Permuted Pre-training for Language Understanding by Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, Tie-Yan Liu.
62. **MT5** (from Google AI) released with the paper mT5: A massively multilingual pre-trained text-to-text transformer by Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, Colin Raffel.
63. **Nyströmformer** (from the University of Wisconsin - Madison) released with the paper Nyströmformer: A Nyström-Based Algorithm for Approximating Self-Attention by Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, Vikas Singh.
64. **Pegasus** (from Google) released with the paper PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization by Jingqing Zhang, Yao Zhao, Mohammad Saleh and Peter J. Liu.
65. **Perceiver IO** (from Deepmind) released with the paper Perceiver IO: A General Architecture for Structured Inputs & Outputs by Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, João Carreira.
66. **PhoBERT** (from VinAI Research) released with the paper PhoBERT: Pre-trained language models for Vietnamese by Dat Quoc Nguyen and Anh Tuan Nguyen.
67. **PLBart** (from UCLA NLP) released with the paper Unified Pre-training for Program Understanding and Generation by Wasi Uddin Ahmad, Saikat Chakraborty, Baishakhi Ray, Kai-Wei Chang.
68. **PoolFormer** (from Sea AI Labs) released with the paper MetaFormer is Actually What You Need for Vision by Yu, Weihao and Luo, Mi and Zhou, Pan and Si, Chenyang and Zhou, Yichen and Wang, Xinchao and Feng, Jiashi and Yan, Shuicheng.
69. **ProphetNet** (from Microsoft Research) released with the paper ProphetNet: Predicting Future N-gram for Sequence-to-Sequence Pre-training by Yu Yan, Weizhen Qi, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang and Ming Zhou.
70. **QDQBert** (from NVIDIA) released with the paper Integer Quantization for Deep Learning Inference: Principles and Empirical Evaluation by Hao Wu, Patrick Judd, Xiaojie Zhang, Mikhail Isaev and Paulius Micikevicius.
71. **REALM** (from Google Research) released with the paper REALM: Retrieval-Augmented Language Model Pre-Training by Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat and Ming-Wei Chang.
72. **Reformer** (from Google Research) released with the paper Reformer: The Efficient Transformer by Nikita Kitaev, Łukasz Kaiser, Anselm Levskaya.
73. **RemBERT** (from Google Research) released with the paper Rethinking embedding coupling in pre-trained language models by Hyung Won Chung, Thibault Févry, Henry Tsai, M. Johnson, Sebastian Ruder.
74. **RegNet** (from META Platforms) released with the paper Designing Network Design Space by Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, Piotr Dollár.
75. **ResNet** (from Microsoft Research) released with the paper Deep Residual Learning for Image Recognition by Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun.
76. **RoBERTa** (from Facebook), released together with the paper RoBERTa: A Robustly Optimized BERT Pretraining Approach by Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov.
77. **RoFormer** (from ZhuiyiTechnology), released together with the paper RoFormer: Enhanced Transformer with Rotary Position Embedding by Jianlin Su and Yu Lu and Shengfeng Pan and Bo Wen and Yunfeng Liu.
78. **SegFormer** (from NVIDIA) released with the paper SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers by Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, Ping Luo.
79. **SEW** (from ASAPP) released with the paper Performance-Efficiency Trade-offs in Unsupervised Pre-training for Speech Recognition by Felix Wu, Kwangyoun Kim, Jing Pan, Kyu Han, Kilian Q. Weinberger, Yoav Artzi.

80. **SEW-D** (from ASAPP) released with the paper Performance-Efficiency Trade-offs in Unsupervised Pre-training for Speech Recognition by Felix Wu, Kwangyoun Kim, Jing Pan, Kyu Han, Kilian Q. Weinberger, Yoav Artzi.
81. **SpeechToTextTransformer** (from Facebook), released together with the paper fairseq S2T: Fast Speech-to-Text Modeling with fairseq by Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, Juan Pino.
82. **SpeechToTextTransformer2** (from Facebook), released together with the paper Large-Scale Self- and Semi-Supervised Learning for Speech Translation by Changhan Wang, Anne Wu, Juan Pino, Alexei Baevski, Michael Auli, Alexis Conneau.
83. **Splinter** (from Tel Aviv University), released together with the paper Few-Shot Question Answering by Pretraining Span Selection by Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, Omer Levy.
84. **SqueezeBert** (from Berkeley) released with the paper SqueezeBERT: What can computer vision teach NLP about efficient neural networks? by Forrest N. Iandola, Albert E. Shaw, Ravi Krishna, and Kurt W. Keutzer.
85. **Swin Transformer** (from Microsoft) released with the paper Swin Transformer: Hierarchical Vision Transformer using Shifted Windows by Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, Baining Guo.
86. **T5** (from Google AI) released with the paper Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer by Colin Raffel and Noam Shazeer and Adam Roberts and Katherine Lee and Sharan Narang and Michael Matena and Yanqi Zhou and Wei Li and Peter J. Liu.
87. **T5v1.1** (from Google AI) released in the repository google-research/text-to-text-transfer-transformer by Colin Raffel and Noam Shazeer and Adam Roberts and Katherine Lee and Sharan Narang and Michael Matena and Yanqi Zhou and Wei Li and Peter J. Liu.
88. **TAPAS** (from Google AI) released with the paper TAPAS: Weakly Supervised Table Parsing via Pre-training by Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno and Julian Martin Eisenschlos.
89. **Transformer-XL** (from Google/CMU) released with the paper Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context by Zihang Dai*, Zhilin Yang*, Yiming Yang, Jaime Carbonell, Quoc V. Le, Ruslan Salakhutdinov.
90. **TrOCR** (from Microsoft), released together with the paper TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models by Minghao Li, Tengchao Lv, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, Furu Wei.
91. **UniSpeech** (from Microsoft Research) released with the paper UniSpeech: Unified Speech Representation Learning with Labeled and Unlabeled Data by Chengyi Wang, Yu Wu, Yao Qian, Kenichi Kumatani, Shujie Liu, Furu Wei, Michael Zeng, Xuedong Huang.
92. **UniSpeechSat** (from Microsoft Research) released with the paper UNISPEECH-SAT: UNIVERSAL SPEECH REPRESENTATION LEARNING WITH SPEAKER AWARE PRE-TRAINING by Sanyuan Chen, Yu Wu, Chengyi Wang, Zhengyang Chen, Zhuo Chen, Shujie Liu, Jian Wu, Yao Qian, Furu Wei, Jinyu Li, Xiangzhan Yu.
93. **VAN** (from Tsinghua University and Nankai University) released with the paper Visual Attention Network by Meng-Hao Guo, Cheng-Ze Lu, Zheng-Ning Liu, Ming-Ming Cheng, Shi-Min Hu.
94. **ViLT** (from NAVER AI Lab/Kakao Enterprise/Kakao Brain) released with the paper ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision by Wonjae Kim, Bokyung Son, Ildoo Kim.
95. **Vision Transformer (ViT)** (from Google AI) released with the paper An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale by Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby.
96. **ViTMAE** (from Meta AI) released with the paper Masked Autoencoders Are Scalable Vision Learners by Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, Ross Girshick.
97. **VisualBERT** (from UCLA NLP) released with the paper VisualBERT: A Simple and Performant Baseline for Vision and Language by Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, Kai-Wei Chang.
98. **WavLM** (from Microsoft Research) released with the paper WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing by Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Furu Wei.
99. **Wav2Vec2** (from Facebook AI) released with the paper wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations by Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, Michael Auli.

100. **Wav2Vec2Phoneme** (from Facebook AI) released with the paper Simple and Effective Zero-shot Cross-lingual Phoneme Recognition by Qiantong Xu, Alexei Baevski, Michael Auli.
101. **XGLM** (From Facebook AI) released with the paper Few-shot Learning with Multilingual Language Models by Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, Xian Li.
102. **XLM** (from Facebook) released together with the paper Cross-lingual Language Model Pretraining by Guillaume Lample and Alexis Conneau.
103. **XLM-ProphetNet** (from Microsoft Research) released with the paper ProphetNet: Predicting Future N-gram for Sequence-to-Sequence Pre-training by Yu Yan, Weizhen Qi, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang and Ming Zhou.
104. **XLM-RoBERTa** (from Facebook AI), released together with the paper Unsupervised Cross-lingual Representation Learning at Scale by Alexis Conneau*, Kartikay Khandelwal*, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer and Veselin Stoyanov.
105. **XLM-RoBERTa-XL** (from Facebook AI), released together with the paper Larger-Scale Transformers for Multilingual Masked Language Modeling by Naman Goyal, Jingfei Du, Myle Ott, Giri Anantharaman, Alexis Conneau.
106. **XLNet** (from Google/CMU) released with the paper XLNet: Generalized Autoregressive Pretraining for Language Understanding by Zhilin Yang*, Zihang Dai*, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le.
107. **XLSR-Wav2Vec2** (from Facebook AI) released with the paper Unsupervised Cross-Lingual Representation Learning For Speech Recognition by Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, Michael Auli.
108. **XLS-R** (from Facebook AI) released with the paper XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale by Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, Michael Auli.
109. **YOSO** (from the University of Wisconsin - Madison) released with the paper You Only Sample (Almost) Once: Linear Cost Self-Attention Via Bernoulli Sampling by Zhanpeng Zeng, Yunyang Xiong, Sathya N. Ravi, Shailesh Acharya, Glenn Fung, Vikas Singh.
110. Want to contribute a new model? We have added a **detailed guide and templates** to guide you in the process of adding a new model. You can find them in the `templates` folder of the repository. Be sure to check the contributing guidelines and contact the maintainers or open an issue to collect feedbacks before starting your PR.

To check if each model has an implementation in Flax, PyTorch or TensorFlow, or has an associated tokenizer backed by the 🤗 Tokenizers library, refer to this table.

These implementations have been tested on several datasets (see the example scripts) and should match the performance of the original implementations. You can find more details on performance in the Examples section of the documentation.

Learn more

Section	Description
Documentation	Full API documentation and tutorials
Task summary	Tasks supported by 🤗 Transformers
Preprocessing tutorial	Using the `Tokenizer` class to prepare data for the models
Training and fine-tuning	Using the models provided by 🤗 Transformers in a PyTorch/TensorFlow training loop and the `Trainer` API
Quick tour: Fine-tuning/usage scripts	Example scripts for fine-tuning models on a wide range of tasks
Model sharing and uploading	Upload and share your fine-tuned models with the community
Migration	Migrate to 🤗 Transformers from `pytorch-transformers` or `pytorch-pretrained-bert`

Citation

We now have a paper you can cite for the 🤗 Transformers library:

```
@inproceedings{wolf-etal-2020-transformers,
    title = "Transformers: State-of-the-Art Natural Language Processing",
    author = "Thomas Wolf and Lysandre Debut and Victor Sanh and Julien Chaumond and Clement Delangue and Anthony Moi and Pierric Cistac and Tim Rault and Rémi Louf and Morgan Funtowicz and Joe Davison and Sam Shleifer and Patrick von Platen and Clara Ma and Yacine Jernite and Julien Plu and Canwen Xu and Teven Le Scao and Sylvain Gugger and Mariama Drame and Quentin Lhoest and Alexander M. Rush",
    booktitle = "Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations",
    month = oct,
    year = "2020",
    address = "Online",
    publisher = "Association for Computational Linguistics",
    url = "https://www.aclweb.org/anthology/2020.emnlp-demos.6",
    pages = "38--45"
}
```