Transfer Learning in NLP with Hugging Face 🤗

Mahdi Samiei, 30 Dec 2020, Sharif University of Technology

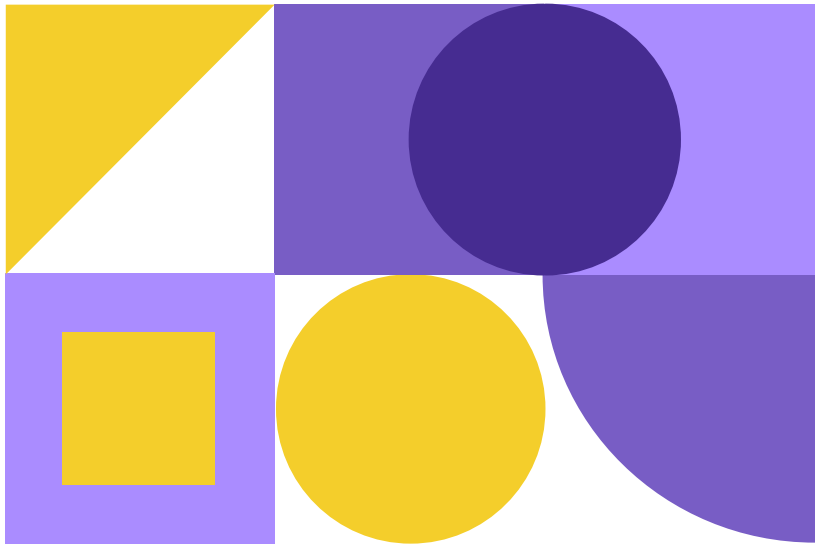
Sharif Winter Seminar Series

Table of contents

01

Theoretical prerequisites

Transformers
Transfer Learning in NLP
Pretrained Models
Bert, GPT, ...



Let's Begin with (generalized) Attention!

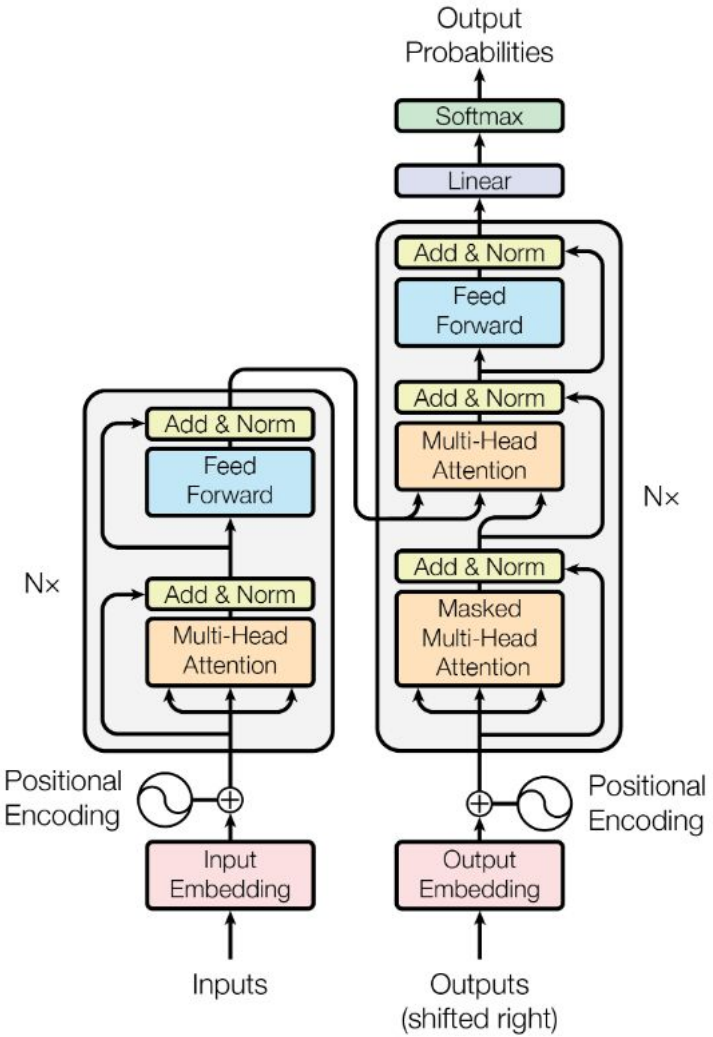
Suppose we some entities: ○ □ △ ⬠

What is Attention of ○ on □ △ ⬠ ?

Every Entity is represented by three vectors:

- Query Vector
- Key Vector
- Value Vector

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



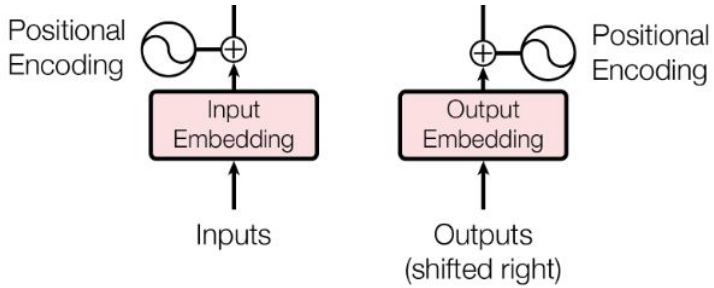
Attention Is All You Need

Transformers Big Revolution in NLP

Other ideas beside attention

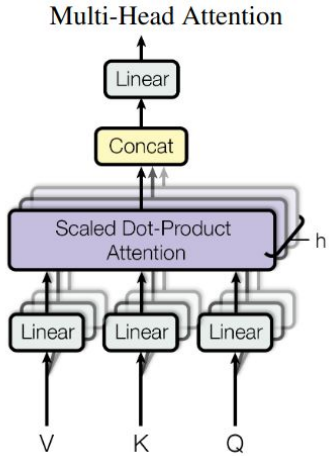
Positional Embedding

To inject positional information about tokens



Multi-Head Attention

To operate more parallel



Advantage and Disadvantages of Transformers

Why we Love Them?

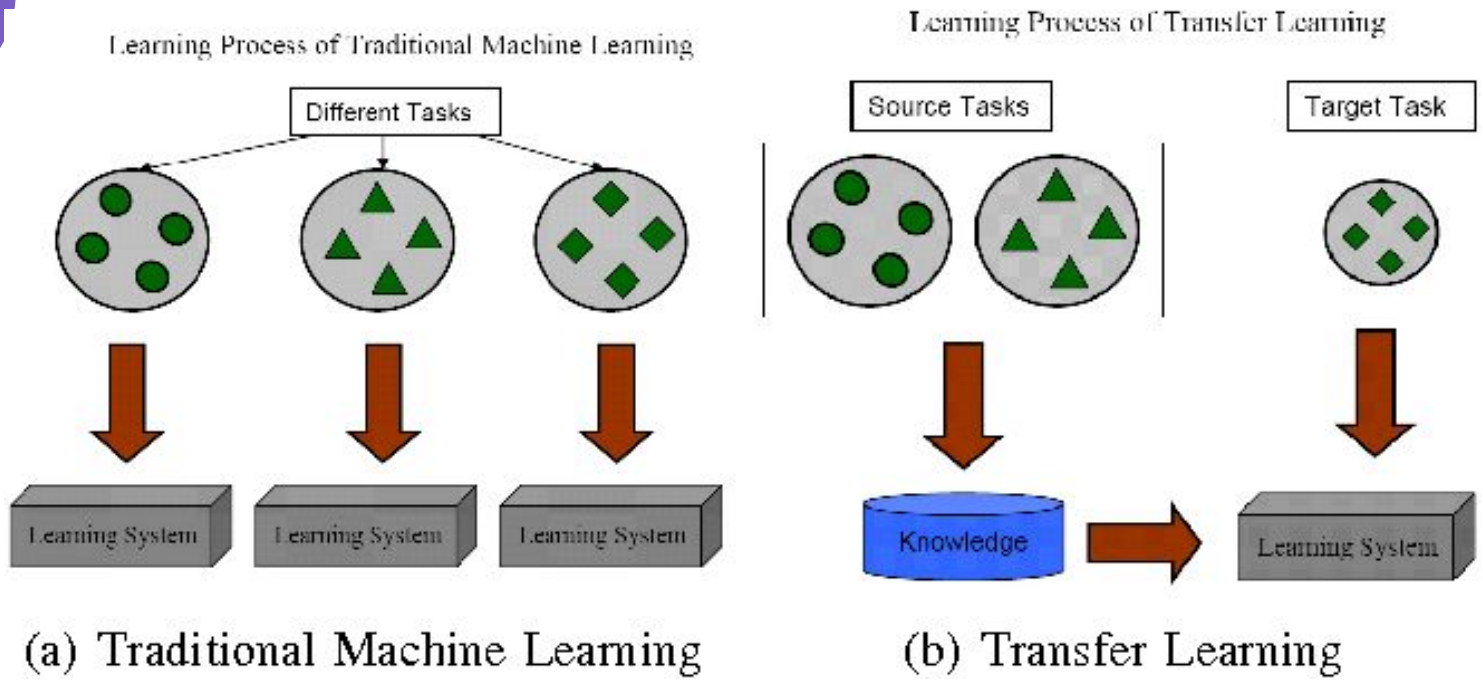
- Parallelism
- No Vanishing!
- Attention Mechanism

Why is it hard to work with them?

- Need more computational resources
- Need more memory

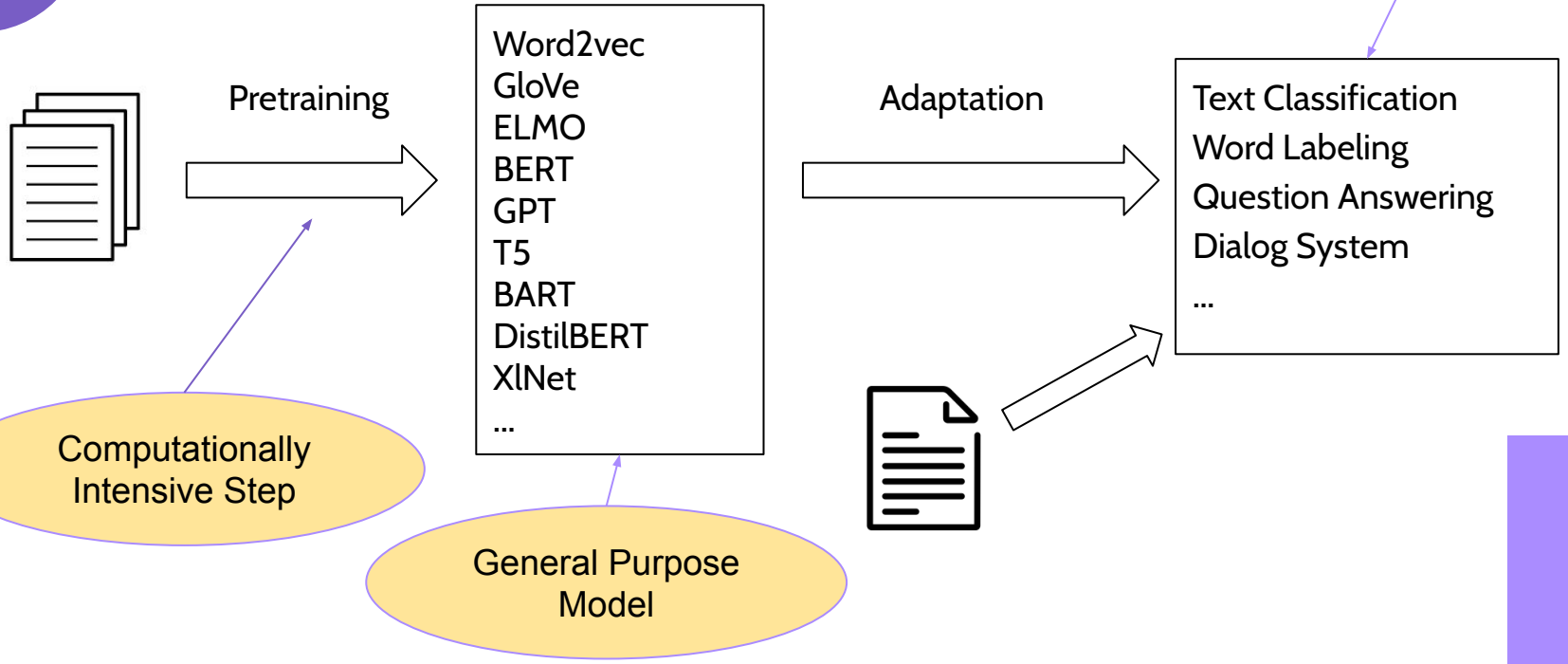
Read more about transformers at https://virgool.io/overfit/

What is Transfer Learning?



Transfer Learning in NLP

Learn on one general task, then transfer it to another more specific task.



Language Modeling Pretraining

- We have snippets of text and want to predict the rest.
  - My house [?????] small.
  - I love my [????].

- learning to predict $P_\theta(text)$ or $P_\theta(text \mid other\ text)$

- Advantages:

  - Doesn’t require human annotation.
  - have enough text.
  - A very general task!

Let’s see two pretraining task

Masker Language Model

In the [MASK] of God.

In the name of God.

Next Sentence Prediction

- The man went to the store.
- He bought a gallon of milk.

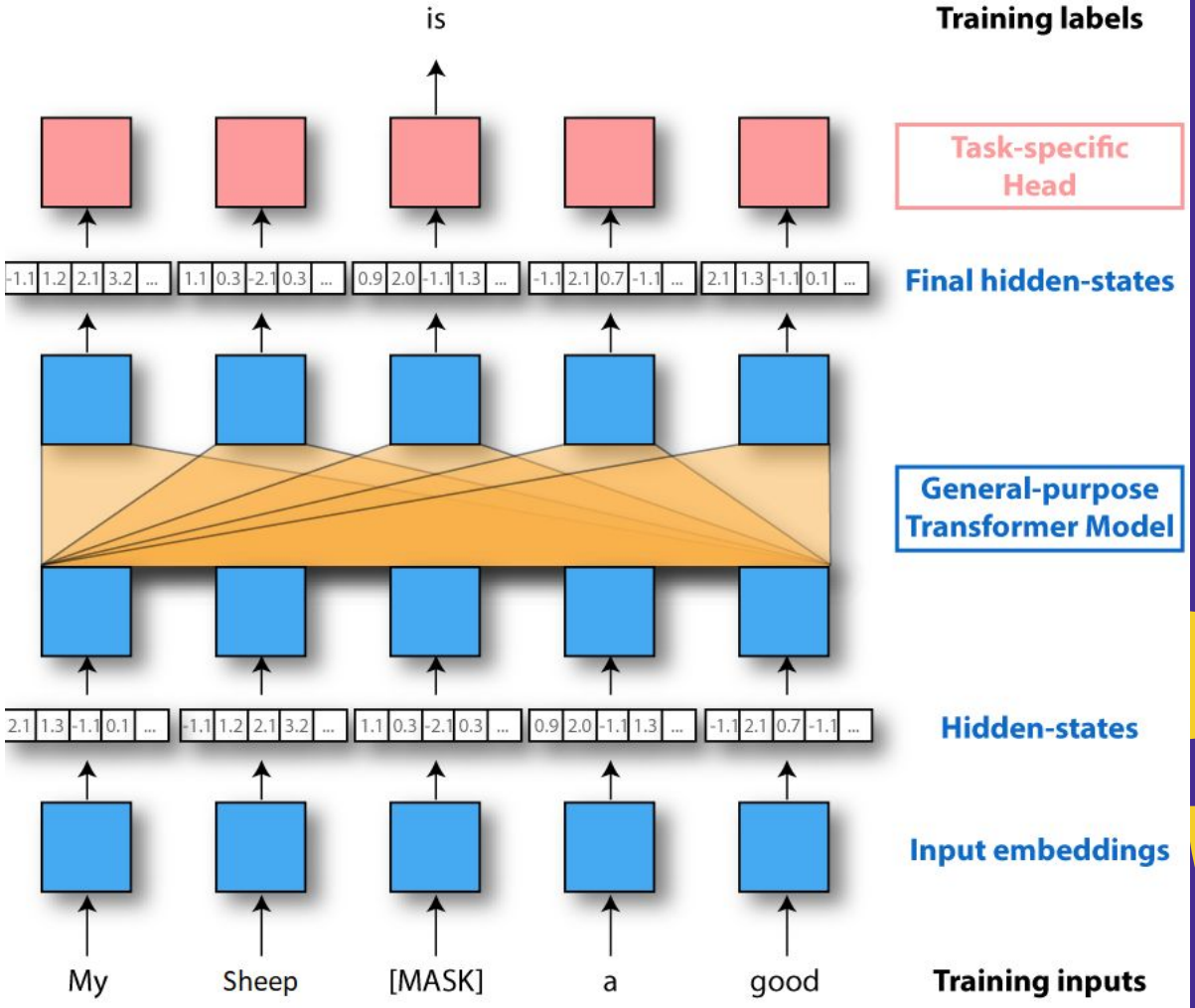
Label: Is Next

- The man went to the store.
- Penguins are flightless birds

Label: Not Next

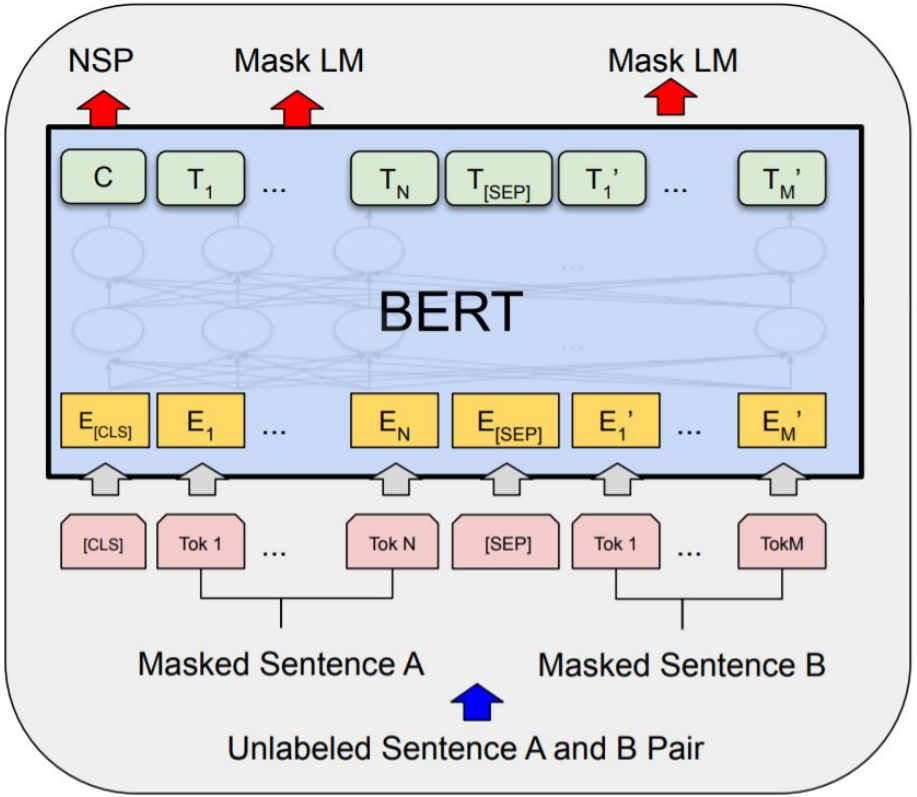
Bert

Solution for mlm



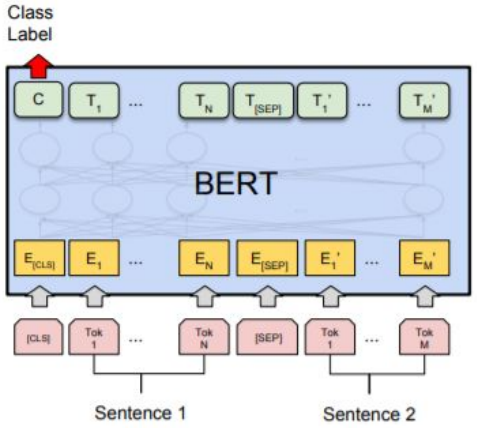
Masked Language Models

Bert Solution For NSP

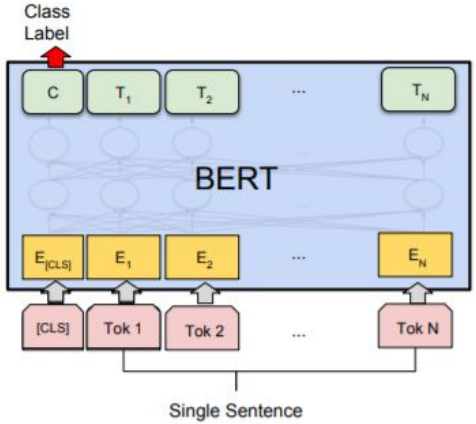


Adaptation is like playing with lego

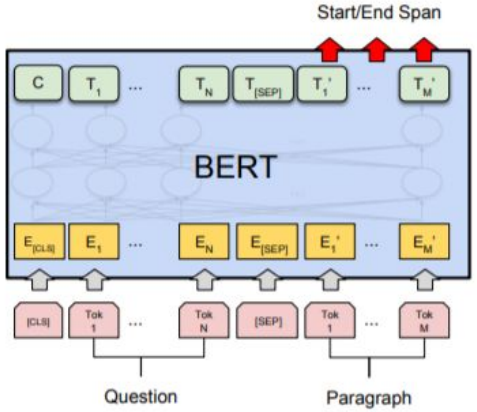
Adaptation Bert to Tasks



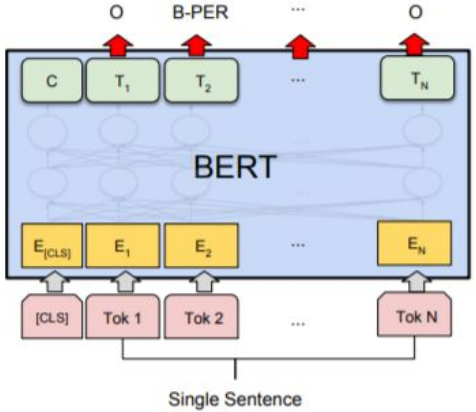
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA



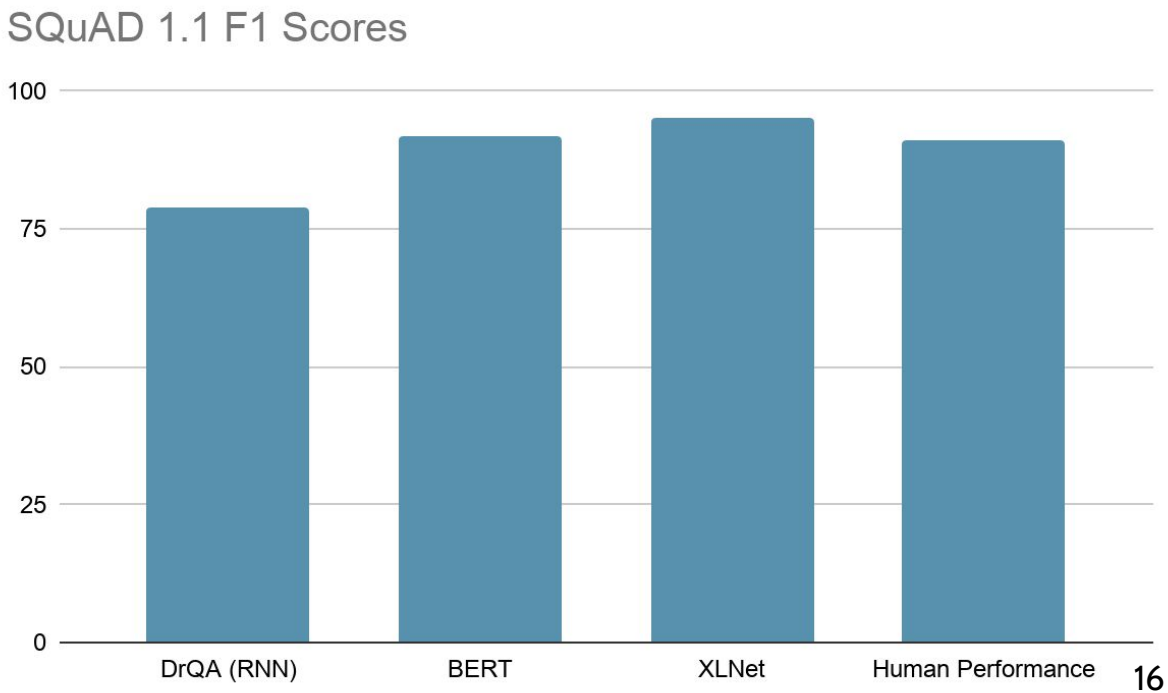
(c) Question Answering Tasks: SQuAD v1.1



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

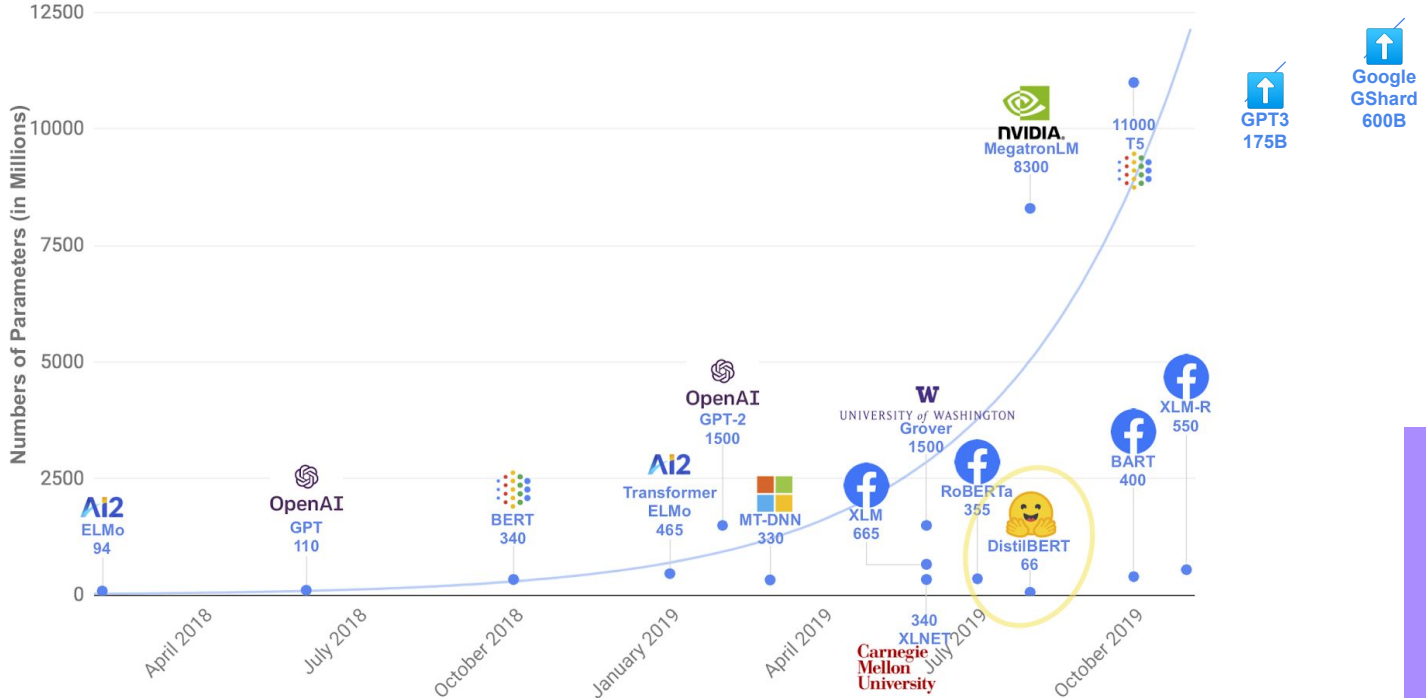
So many Pretrained Models ...

- OpenAI GPT, GPT2, GPT3
- Roberta
- Albert
- BART
- T5
- Reformer
- Big Bird
- ...



But WAIT! Where is NLP going?

- Narrowing the research competition field.
- Difficulties of training and deployment.

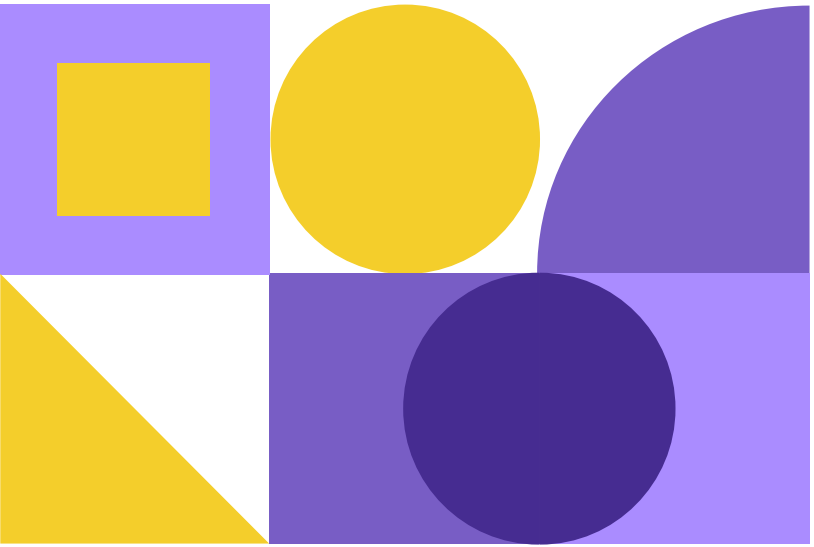


Some new trends

Reducing the size of a pretrained model

Three main techniques currently investigated:

- Distillation
  - DistilBert: 95% of Bert performances in a model 40% smaller and 60% faster
- Pruning
- Quantization
  - From FP32 to INT8



Hugging Face

Hugging Face: Democratizing NLP

- Develop & open-source tools for Transfer Learning in NLP
- to **accelerate**, **catalyse** and **democratize** research-level work in Natural Language Understanding as well as Natural Language Generation
- Code & model sharing: Open-sourcing the “right way”
  - In the middle of 1000-commands research-code ⇔ 1-command production code
  - PyTorch / TensorFlow / Jax
  - Make people **stand on the shoulders of giants**

Transformers Library

- a library dedicated to supporting Transformer-based architectures and facilitating the distribution of pretrained models.
- supports the distribution and usage of a wide-variety of pretrained models in a centralized model hub
- a vibrant community of over 400 external contributors.

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. 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. DeBERTa (from Microsoft Research) released with the paper DeBERTa: Decoding-enhanced BERT with Disentangled Attention by Pengcheng He, Xiaodong Liu, Jianfeng Gao, Weizhu Chen.
10. 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.
11. 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.
12. 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.
13. 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.
14. 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.
15. 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.
16. GPT (from OpenAI) released with the paper Improving Language Understanding by Generative Pre-Training by Alec Radford, Karthik Narasimhan, Tim Salimans and Ilya Sutskever.
17. 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**.
18. 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.
19. Longformer (from AllenAI) released with the paper Longformer: The Long-Document Transformer by Iz Beltagy, Matthew E. Peters, Arman Cohan.
20. 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.
21. MarianMT Machine translation models trained using OPUS data by Jörg Tiedemann. The Marian Framework is being developed by the Microsoft Translator Team.
22. 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.
23. 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.
24. 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.
25. 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.
26. 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.
27. Reformer (from Google Research) released with the paper Reformer: The Efficient Transformer by Nikita Kitaev, Łukasz Kaiser, Anselm Levskaya.
28. RoBERTa (from Facebook), released together with the paper 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. ultilingual BERT into DistilmBERT and a German version of DistilBERT.
29. SqueezeBert 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.
30. 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.
31. 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.
32. 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.
33. XLM (from Facebook) released together with the paper Cross-lingual Language Model Pretraining by Guillaume Lample and Alexis Conneau.
34. 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.
35. 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.
36. 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.

Transformers library: code example

```python
import torch
from transformers import *

# Transformers has a unified API
# for 8 transformer architectures and 30 pretrained weights.
#          Model          | Tokenizer          | Pretrained weights shortcut
MODELS = [(BertModel,       BertTokenizer,       'bert-base-uncased'),
          (OpenAIGPTModel,  OpenAIGPTTokenizer,  'openai-gpt'),
          (GPT2Model,       GPT2Tokenizer,       'gpt2'),
          (TransfoXLModel,  TransfoXLTokenizer,  'transfo-xl-wt103'),
          (XLNetModel,      XLNetTokenizer,      'xlnet-base-cased'),
          (XLMModel,        XLMTokenizer,        'xlm-mlm-enfr-1024'),
          (DistilBertModel, DistilBertTokenizer, 'distilbert-base-uncased'),
          (RobertaModel,    RobertaTokenizer,    'roberta-base')]

# To use TensorFlow 2.0 versions of the models, simply prefix the class names with 'TF', e.g. `TFRobertaM

# Let's encode some text in a sequence of hidden-states using each model:
for model_class, tokenizer_class, pretrained_weights in MODELS:
    # Load pretrained model/tokenizer
    tokenizer = tokenizer_class.from_pretrained(pretrained_weights)
    model = model_class.from_pretrained(pretrained_weights)

    # Encode text
    input_ids = torch.tensor([tokenizer.encode("Here is some text to encode", add_special_tokens=True)])
    with torch.no_grad():
        last_hidden_states = model(input_ids)[0]  # Models outputs are now tuples

# Each architecture is provided with several class for fine-tuning on down-stream tasks, e.g.
BERT_MODEL_CLASSES = [BertModel, BertForPreTraining, BertForMaskedLM, BertForNextSentencePrediction,
                      BertForSequenceClassification, BertForMultipleChoice, BertForTokenClassification,
                      BertForQuestionAnswering]
```

Transformers library: code example

```
import torch
from transformers import *

# Transformers has a unified API
# for 8 transformer architectures and 30 pretrained weights.
#          Model          | Tokenizer          | Pretrained weights shortcut
MODELS = [(BertModel,       BertTokenizer,       'bert-base-uncased'),
          (OpenAIGPTModel,  OpenAIGPTTokenizer,  'openai-gpt'),
          (GPT2Model,       GPT2Tokenizer,       'gpt2'),
          (TransfoXLModel,  TransfoXLTokenizer,  'transfo-xl-wt103'),
          (XLNetModel,      XLNetTokenizer,      'xlnet-base-cased'),
          (XLMModel,        XLMTokenizer,        'xlm-mlm-enfr-1024'),
          (DistilBertModel, DistilBertTokenizer, 'distilbert-base-uncased'),
          (RobertaModel,    RobertaTokenizer,    'roberta-base')]

    # Encode text
    input_ids = torch.tensor([tokenizer.encode("Here is some text to encode", add_special_tokens=True)])
    with torch.no_grad():
        last_hidden_states = model(input_ids)[0]  # Models outputs are now tuples

# Each architecture is provided with several class for fine-tuning on down-stream tasks, e.g.
BERT_MODEL_CLASSES = [BertModel, BertForPreTraining, BertForMaskedLM, BertForNextSentencePrediction,
                      BertForSequenceClassification, BertForMultipleChoice, BertForTokenClassification,
                      BertForQuestionAnswering]
```

💥 Check it out at 💥
https://github.com/huggingface/transformers

Transformers: model hub

All Models and checkpoints

Also check out our list of **Community contributors** 🏆 and **Organizations** 🌍.

Search models... | Tags: All | Sort: Most downloads

- bert-base-multilingual-cased
- distilbert-base-uncased
- jplu/tf-xlm-roberta-base
- bert-base-uncased
- bert-base-cased
- gpt2
- distilbert-base-cased-distilled-squad
- roberta-large
- roberta-base
- facebook/bart-large-cnn
- bert-large-uncased
- xlm-roberta-large-finetuned-conll03-german
- distilbert-base-cased
- distilroberta-base
- deepset/roberta-base-squad2
- SpanBERT/spanbert-large-cased
- etalab-ia/camembert-base-squadFR-fquad-piaf
- microsoft/DialoGPT-large

🌍 All Languages 🌍

This table displays the number of **mono-lingual** (or "few"-lingual, with "few" arbitrarily set to 5 or less) models, by language. You can click on the figures on the right to the list of actual models. **Multilingual** models are listed **here** 🌍.

Language	ISO code	Num. of models
English	en	465
Spanish	es	231
French	fr	216
Swedish	sv	203
Finnish	fi	181
German	de	132
Russian	ru	59
Ukrainian	uk	54
Italian	it	52
Esperanto	eo	50
Arabic	ar	43
Bulgarian	bg	39
Polish	pl	37
Dutch	nl	36
Chinese	zh	35
Turkish	tr	34
Afrikaans	af	30
Catalan	ca	28
Czech	cs	27
Norwegian	no	26
Indonesian	id	26
Hebrew	he	26
Danish	da	26
Portuguese	pt	25
Icelandic	is	25

HUGGING FACE

Back to all models

Model: `bert-base-multilingual-cased`

pytorch | tf | bert | lm-head | masked-lm | multilingual | dataset:wikipedia | license:apache-2.0

Hosted inference API

fill-mask — mask_token: [MASK]

Your sentence here...

Compute

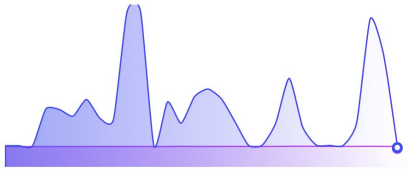
🔥 This model is currently loaded and running on the Inference API.

Monthly model downloads

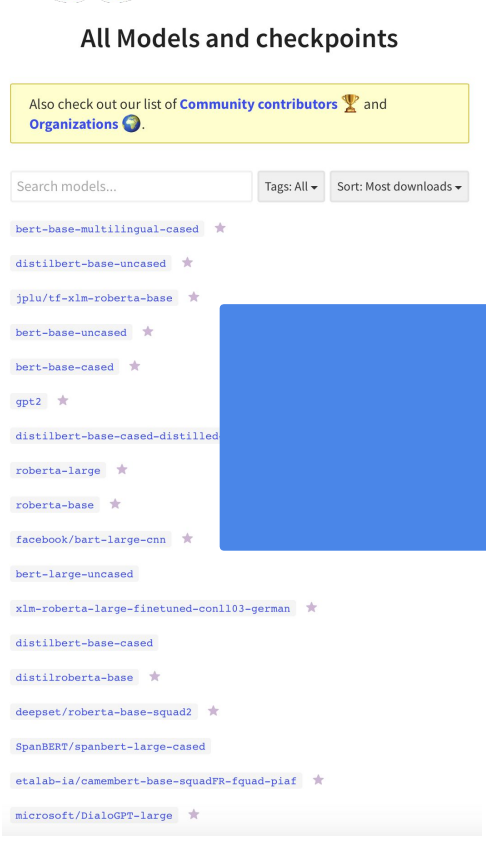
bert-base-multilingual-cased

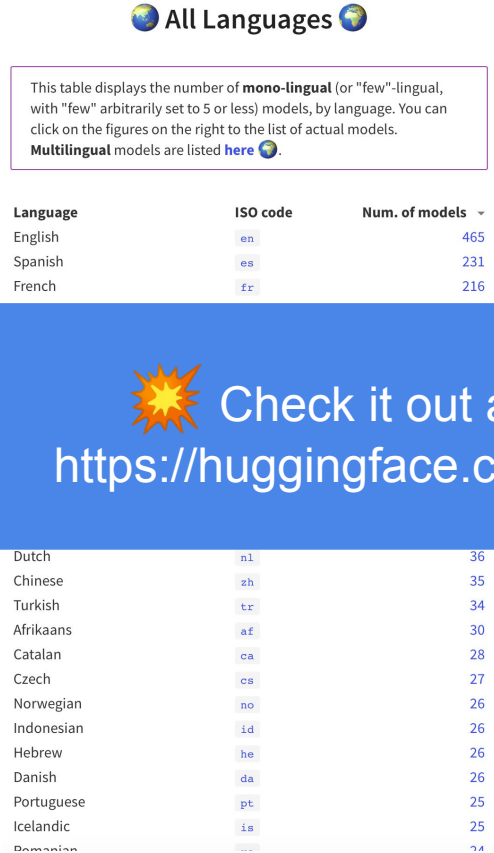
19,904,557 downloads last 30 days

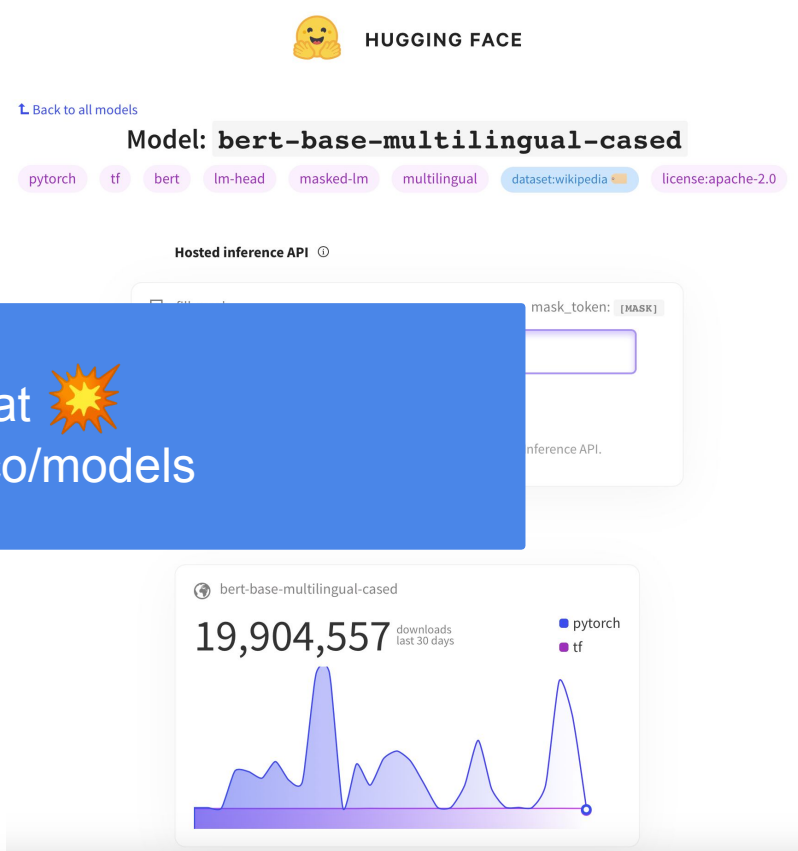
pytorch
tf



Transformers: model hub







Check it out at
https://huggingface.co/models

🤗 Tokenizers library

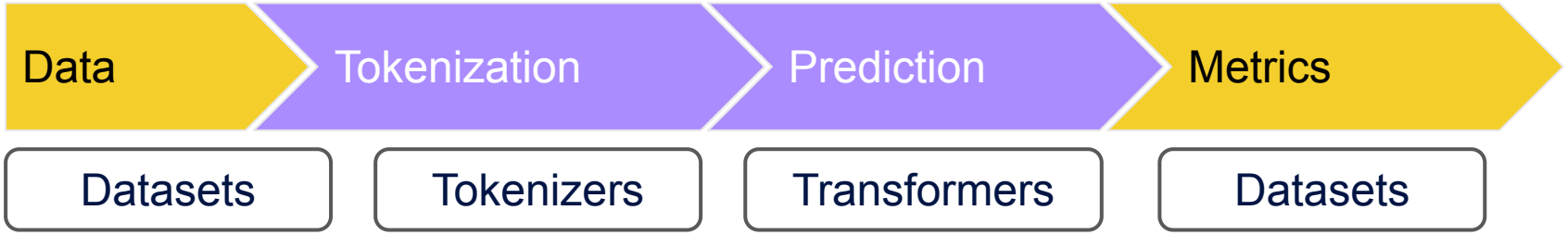
Now that neural nets have fast implementations, a **bottleneck** in Deep-Learning based NLP pipelines is often **tokenization**: converting strings ➡️ model inputs.

🤗Tokenizers: **ultra-fast & versatile tokenization**

Features:

1. Encode **1GB** in **20sec**
2. BPE/byte-level-BPE/WordPiece/SentencePiece...
3. Bindings in **python/js/rust...**
4. Link: https://github.com/huggingface/tokenizers

Datasets library



- The full data-processing pipeline goes beyond tokenization and models to include data access and preprocessing at the beginning and model evaluation at the end.
- a new library 🤗**Datasets** to improve the situation on both ends of the pipeline.
- a lightweight and extensible library to easily access and process datasets and evaluation metrics for Natural Language Processing (NLP).

Datasets: code example

```
from nlp import load_dataset, load_metric
from transformers import AutoTokenizer, AutoModelForSequenceClassification, AdamW
from torch.utils.data import DataLoader

dataset = load_dataset('glue', 'mrpc')
metric = load_metric('glue', 'mrpc')
tokenizer = AutoTokenizer.from_pretrained('bert-base-cased')

# Encode our dataset
def encode(example):
  output = tokenizer(example['sentence1'], example['sentence2'], truncation=True)
  output['labels'] = example['label']
  return output

dataset = dataset.map(encode)
dataset.set_format(columns=['attention_mask', 'input_ids', 'token_type_ids', 'labels'])

# Prepare pytorch dataloader (with dynamic batch i.e. pad the sequences to the longest in the batch)
def collate_fn(examples):
  return tokenizer.pad(examples, padding='longest', return_tensors='pt')

train_dataloader = DataLoader(dataset['train'], shuffle=True, collate_fn=collate_fn, batch_size=32)
valid_dataloader = DataLoader(dataset['validation'], shuffle=False, collate_fn=collate_fn, batch_size=16)

# We are ready for training and evaluating a PyTorch model!
```

Datasets: code example

```
from nlp import load_dataset, load_metric
from transformers import AutoTokenizer, AutoModelForSequenceClassification, AdamW
from torch.utils.data import DataLoader

dataset = load_dataset('glue', 'mrpc')
metric = load_metric('glue', 'mrpc')
tokenizer = AutoTokenizer.from_pretrained('bert-base-cased')

# Enco
def en
  outp
  outp
  retu

datase
datase

# Prepare pytorch dataloader (with dynamic batch i.e. pad the sequences to the longest in the batch)
def collate_fn(examples):
  return tokenizer.pad(examples, padding='longest', return_tensors='pt')

train_dataloader = DataLoader(dataset['train'], shuffle=True, collate_fn=collate_fn, batch_size=32)
valid_dataloader = DataLoader(dataset['validation'], shuffle=False, collate_fn=collate_fn, batch_size=16)

# We are ready for training and evaluating a PyTorch model!
```

💥 Check it out at 💥
https://github.com/huggingface/datasets

Datasets: datasets hub

Back to home

All Datasets

All datasets from our `nlp` repository and community bucket.
Also check out the list of supported **Metrics**.

Search datasets... | Sort: Default

aeslc — 1 model

A collection of email messages of employees in the Enron Corporation. There are two features: - email_body: email body text. - subject_line: email subject text.

ag_news

AG is a collection of more than 1 million news articles. News articles have been gathered from more than 2000 news sources by ComeToMyHead in more than 1 year of activity. ComeToMyHead is an academic news search engine which has been running since July, 2004. The dataset is provided b…

ai2_arc

A new dataset of 7,787 genuine grade-school level, multiple-choice science questions, assembled to encourage research in advanced question-answering. The dataset is partitioned into a Challenge Set and an Easy Set, where the former contains only questions answered incorrectly by both a…

allocine

Allocine Dataset: A Large-Scale French Movie Reviews Dataset. This is a dataset for binary sentiment classification, made of user reviews scraped from Allocine.fr. It contains 100k positive and 100k negative reviews divided into 3 balanced splits: train (160k reviews), val (20k) and test (20k).

anli

The Adversarial Natural Language Inference (ANLI) is a new large-scale NLI benchmark dataset, The dataset is collected via an iterative, adversarial

Back to all datasets

Dataset: bookcorpus

Update on GitHub

Browse nlp/viewer

How to load this dataset directly with the `nlp` library:

```
from nlp import load_dataset

dataset = load_dataset("bookcorpus")
```

Description

Books are a rich source of both fine-grained information, how a character, an object or a scene looks like, as well as high-level semantics, what someone is thinking, feeling and how these states evolve through a story.This work aims to align books to their movie releases in order to providerich descriptive explanations for visual content that go semantically farbeyond the captions available in current datasets. \

Citation

```
@InProceedings{Zhu_2015_ICCV,
    title = {Aligning Books and Movies: Towards Story-Like Visual Exp
    author = {Zhu, Yukun and Kiros, Ryan and Zemel, Rich and Salakhut
    booktitle = {The IEEE International Conference on Computer Vision
    month = {December},
    year = {2015}
}
```

Models trained or fine-tuned on bookcorpus

`bert-base-cased` ★

`bert-base-uncased` ★

`distilbert-base-uncased` ★

`roberta-base` ★

nlp

https://github.com/huggingface/nlp

Docs | Overview | Add Dataset

Dataset: bookcorpus

Split: train

Offset (Size: 74004228): 0

Show Citations

Show List View

Show Features

Code

```
!pip install nlp
from nlp import load_dataset
dataset = load_dataset(
    'bookcorpus')
```

Dataset: bookcorpus

Homepage: https://yknzhu.wixsite.com/mbweb

Dataset: https://github.com/huggingface/nlp/blob/master/datasets/bookcorpus/bookcorpus.py

Books are a rich source of both fine-grained information, how a character, an object or a scene looks like, as well as high-level semantics, what someone is thinking, feeling and how these states evolve through a story.This work aims to align books to their movie releases in order to providerich descriptive explanations for visual content that go semantically farbeyond the captions available in current datasets.

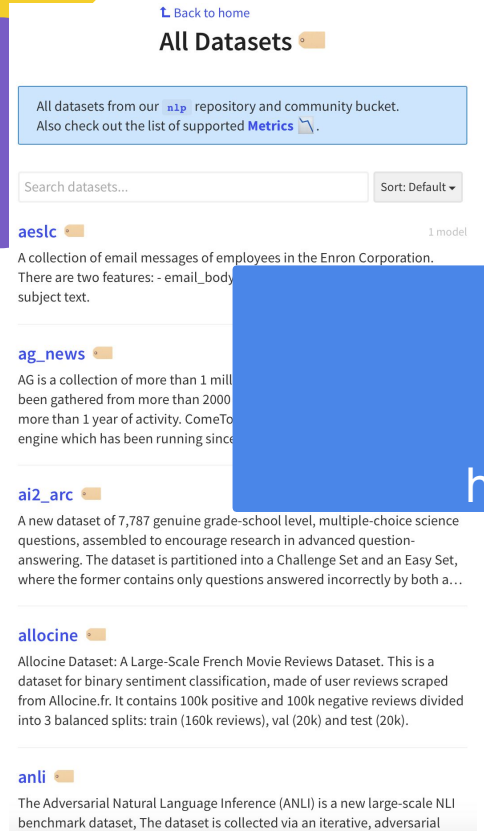
Features

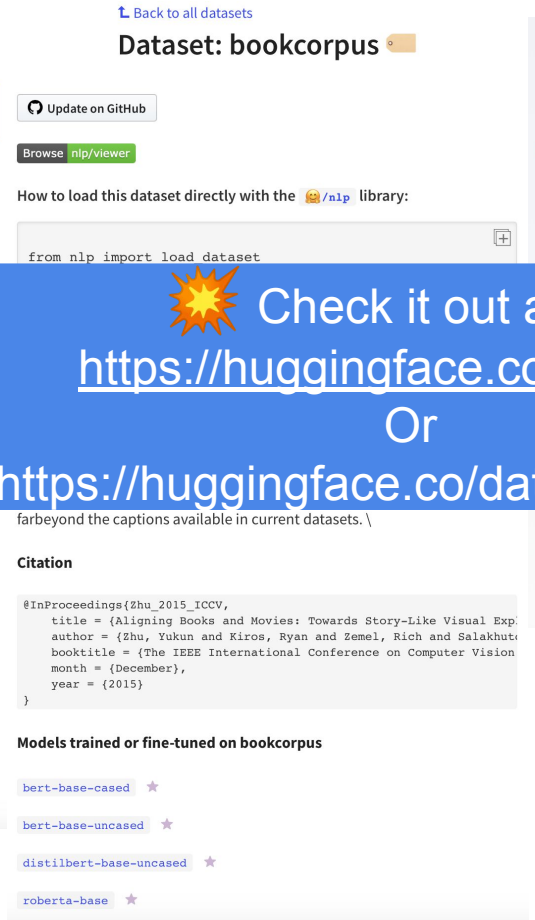
text

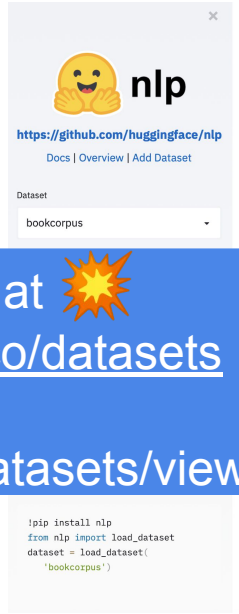
```
{
  "text" : "string"
}
```

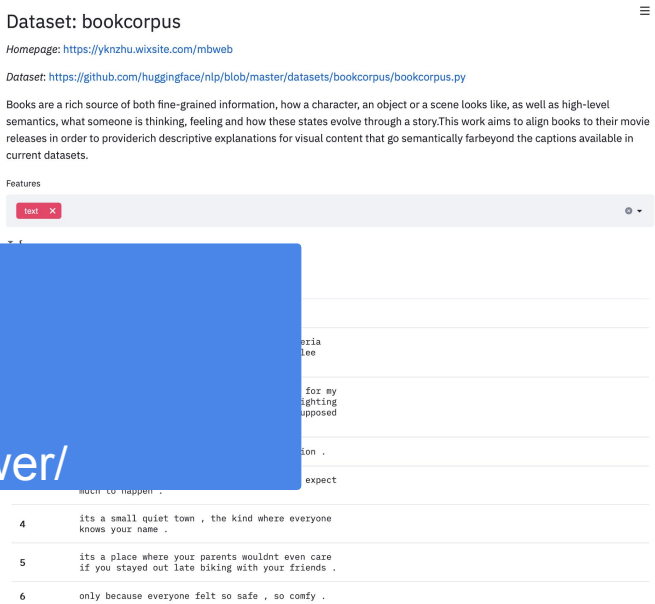
	text
0	the half-ling book one in the fall of igneeria series kaylee soderburg copyright 2013 kaylee soderburg all rights reserved .
1	isbn : 1492913731 isbn-13 : 978-1492913733 for my family , who encouraged me to never stop fighting for my dreams chapter 1 summer vacations supposed to be fun , right ?
2	i wish i had a better answer to that question .
3	starlings , new york is not the place youd expect much to happen .
4	its a small quiet town , the kind where everyone knows your name .
5	its a place where your parents wouldnt even care if you stayed out late biking with your friends .
6	only because everyone felt so safe , so comfy .

Datasets: datasets hub









Check it out at
https://huggingface.co/datasets
Or
https://huggingface.co/datasets/viewer/

Hosted Inference API: NLP as a Service

HUGGING FACE

Back to home

Pricing

Plans for community and organizations

Community

Contributor
Free

- **Upload public models**
- Community support
- Follow tags for new model alerts
- Soon: compare models with model hub history

Create a free account

Become a Supporter
$9/mo

- **Get a PRO badge on the website and the forum**
  Show your support for the best community in ML
- **Upload up to 5 private models**
- Community support
- Early preview access to all the upcoming hub features
- Soon: compare models with model hub history

Start now

Organizations

Lab
$199/mo

- **Upload private models**
- **Access to hosted Inference API: Accelerated CPU**
  2x faster than inference widgets.
- 1 preloaded model
- 10M input characters
- Email support

Start 7-day free trial

Startup
$599/mo

- **Upload private models**
- **Access to hosted Inference API: Accelerated CPU + GPU**
  We pick the best hardware for your models.
- 5 preloaded models
- 50M input characters
- Email support

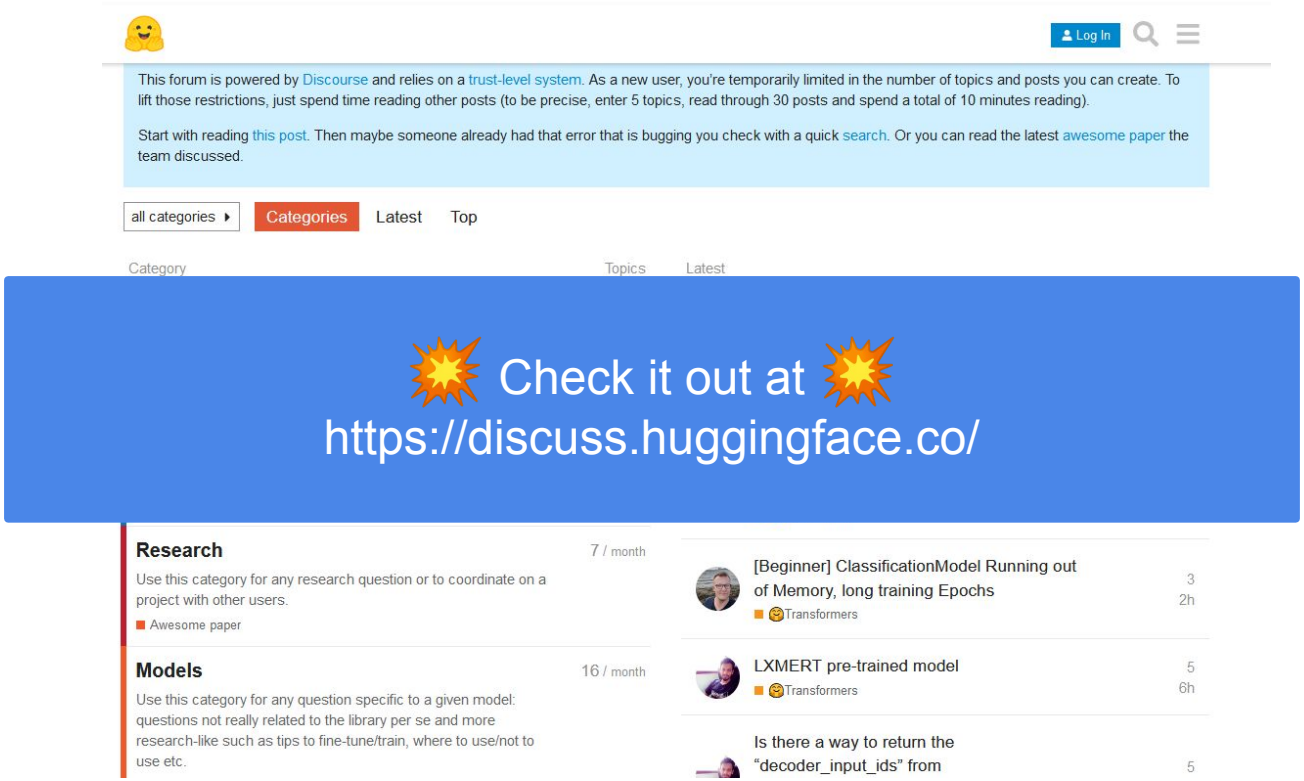
Start 7-day free trial

Enterprise
Custom quote

- **We will design a custom solution to your requirements**
  If your company is building its own infrastructure, our Expert Acceleration Program can support your team, and our Containerized Solutions can be deployed on-premise.
- **Priority support** and access to the Hugging Face team

Contact us

Hugging Face Forums



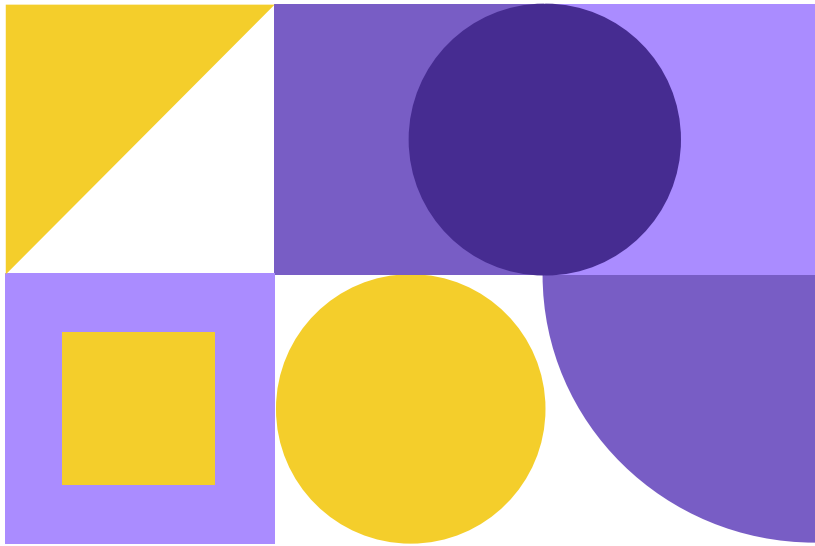
💥 Check it out at 💥
https://discuss.huggingface.co/

Resources

- Wolf, Thomas. “An introduction to transfer learning in NLP and HuggingFace with Thomas Wolf” YouTube, uploaded by MLT Artificial Intelligence, 30 Oct. 2020, https://www.youtube.com/watch?v=t86G11tfVNw
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin,“Attention is all you need,” in Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017
- J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” CoRR, vol. abs/1810.04805, 2018.

Thank you!

we'll be very glad to see you at
t.me/nlp_stuff



Questions?

For more questions:
mohmahsamiei@gmail.com