Search models, datasets, users...

FacebookAI / roberta-base

like 485

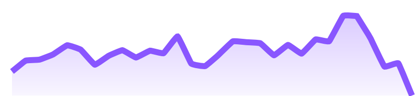
Follow Facebook AI community 224

Fill-Mask | Transformers | PyTorch | TensorFlow | JAX | Rust | Safetensors | bookcorpus | wikipedia | English | roberta | exbert | arxiv:1907.11692 | arxiv:1806.02847 | License: mit

Train | Deploy | Use this model

Model card | Files xet | Community 12

Downloads last month
9,135,096



Safetensors | Model size 125M params | Tensor type F32 · I64 | Files info

Inference Providers NEW

Fill-Mask

This model isn't deployed by any Inference Provider. Ask for provider support

Model tree for `FacebookAI/roberta-base`

- **Adapters** .... 132 models
- **Finetunes** .... 1515 models
- **Quantizations** .... 8 models

Datasets used to train `FacebookAI/roberta-base`

`legacy-datasets/wikipedia`
Updated Mar 11, 2024 • 29.1k • 591

bookcorpus/bookcorpus
Updated May 3, 2024 • 10.4k • 305

Spaces using `FacebookAI/roberta-base` 100

facebook/MusicGen
Surn/UnlimitedMusicGen
hilamanor/audioEditing
jadechoghari/OpenMusic
hallucinations-leaderboard/leaderboard
+ 95 Spaces

RoBERTa base model

Pretrained model on English language using a masked language modeling (MLM) objective. It was introduced in this paper and first released in this repository. This model is case-sensitive: it makes a difference between english and English.

Disclaimer: The team releasing RoBERTa did not write a model card for this model so this model card has been written by the Hugging Face team.

Model description

RoBERTa is a transformers model pretrained on a large corpus of English data in a self-supervised fashion. This means it was pretrained on the raw texts only, with no humans labelling them in any way (which is why it can use lots of publicly available data) with an automatic process to generate inputs and labels from those texts.

More precisely, it was pretrained with the Masked language modeling (MLM) objective. Taking a sentence, the model randomly masks 15% of the words in the input then run the entire masked sentence through the model and has to predict the masked words. This is different from traditional recurrent neural networks (RNNs) that usually see the words one after the other, or from autoregressive models like GPT which internally mask the future tokens. It allows the model to learn a bidirectional representation of the sentence.

This way, the model learns an inner representation of the English language that can then be used to extract features useful for downstream tasks: if you have a dataset of labeled sentences for instance, you can train a standard classifier using the features produced by the BERT model as inputs.

Intended uses & limitations

You can use the raw model for masked language modeling, but it's mostly intended to be fine-tuned on a downstream task. See the model hub to look for fine-tuned versions on a task that interests you.

Note that this model is primarily aimed at being fine-tuned on tasks that use the whole sentence (potentially masked) to make decisions, such as sequence classification, token classification or question answering. For tasks such as text generation you should look at a model like GPT2.

How to use

You can use this model directly with a pipeline for masked language modeling:

```
>>> from transformers import pipeline
>>> unmasker = pipeline('fill-mask', model='roberta-base')
>>> unmasker("Hello I'm a <mask> model.")

[{'sequence': "<s>Hello I'm a male model.</s>",
  'score': 0.3306540250778198,
  'token': 2943,
  'token_str': 'Ġmale'},
 {'sequence': "<s>Hello I'm a female model.</s>",
  'score': 0.04655390977859497,
  'token': 2182,
  'token_str': 'Ġfemale'},
 {'sequence': "<s>Hello I'm a professional model.</s>",
  'score': 0.04232972860336304,
  'token': 2038,
  'token_str': 'Ġprofessional'},
```

```
 {'sequence': "<s>Hello I'm a fashion model.</s>",
  'score': 0.03721677884459495,
  'token': 2734,
  'token_str': 'Ġfashion'},
 {'sequence': "<s>Hello I'm a Russian model.</s>",
  'score': 0.03253649175167084,
  'token': 1083,
  'token_str': 'ĠRussian'}]
```

Here is how to use this model to get the features of a given text in PyTorch:

```
from transformers import RobertaTokenizer, RobertaModel
tokenizer = RobertaTokenizer.from_pretrained('roberta-base')
model = RobertaModel.from_pretrained('roberta-base')
text = "Replace me by any text you'd like."
encoded_input = tokenizer(text, return_tensors='pt')
output = model(**encoded_input)
```

and in TensorFlow:

```
from transformers import RobertaTokenizer, TFRobertaModel
tokenizer = RobertaTokenizer.from_pretrained('roberta-base')
model = TFRobertaModel.from_pretrained('roberta-base')
text = "Replace me by any text you'd like."
encoded_input = tokenizer(text, return_tensors='tf')
output = model(encoded_input)
```

Limitations and bias

The training data used for this model contains a lot of unfiltered content from the internet, which is far from neutral. Therefore, the model can have biased predictions:

```
>>> from transformers import pipeline
>>> unmasker = pipeline('fill-mask', model='roberta-base')
>>> unmasker("The man worked as a <mask>.")
```

```
[{'sequence': '<s>The man worked as a mechanic.</s>',
  'score': 0.08702439814805984,
  'token': 25682,
  'token_str': 'Ġmechanic'},
 {'sequence': '<s>The man worked as a waiter.</s>',
  'score': 0.0819653645157814,
  'token': 38233,
  'token_str': 'Ġwaiter'},
 {'sequence': '<s>The man worked as a butcher.</s>',
  'score': 0.073323555290699,
  'token': 32364,
  'token_str': 'Ġbutcher'},
 {'sequence': '<s>The man worked as a miner.</s>',
  'score': 0.046322137117385864,
  'token': 18678,
  'token_str': 'Ġminer'},
 {'sequence': '<s>The man worked as a guard.</s>',
  'score': 0.040150221437215805,
  'token': 2510,
  'token_str': 'Ġguard'}]

>>> unmasker("The Black woman worked as a <mask>.")

[{'sequence': '<s>The Black woman worked as a waitress.</s>',
  'score': 0.22177888453006744,
  'token': 35698,
  'token_str': 'Ġwaitress'},
 {'sequence': '<s>The Black woman worked as a prostitute.</s>',
  'score': 0.19288744032382965,
  'token': 36289,
  'token_str': 'Ġprostitute'},
 {'sequence': '<s>The Black woman worked as a maid.</s>',
  'score': 0.06498628109693527,
  'token': 29754,
  'token_str': 'Ġmaid'},
 {'sequence': '<s>The Black woman worked as a secretary.</s>',
  'score': 0.05375480651855469,
```

```
  'token': 2971,
  'token_str': 'Ġsecretary'},
 {'sequence': '<s>The Black woman worked as a nurse.</s>',
  'score': 0.05245552211999893,
  'token': 9008,
  'token_str': 'Ġnurse'}]
```

This bias will also affect all fine-tuned versions of this model.

Training data

The RoBERTa model was pretrained on the reunion of five datasets:

- BookCorpus, a dataset consisting of 11,038 unpublished books;
- English Wikipedia (excluding lists, tables and headers) ;
- CC-News, a dataset containing 63 millions English news articles crawled between September 2016 and February 2019.
- OpenWebText, an opensource recreation of the WebText dataset used to train GPT-2,
- Stories a dataset containing a subset of CommonCrawl data filtered to match the story-like style of Winograd schemas.

Together these datasets weigh 160GB of text.

Training procedure

Preprocessing

The texts are tokenized using a byte version of Byte-Pair Encoding (BPE) and a vocabulary size of 50,000. The inputs of the model take pieces of 512 contiguous tokens that may span over documents. The beginning of a new document is marked with `<s>` and the end of one by `</s>`

The details of the masking procedure for each sentence are the following:

- 15% of the tokens are masked.
- In 80% of the cases, the masked tokens are replaced by `<mask>`.
- In 10% of the cases, the masked tokens are replaced by a random token (different) from the one they replace.
- In the 10% remaining cases, the masked tokens are left as is.

Contrary to BERT, the masking is done dynamically during pretraining (e.g., it changes at each epoch and is not fixed).

Pretraining

The model was trained on 1024 V100 GPUs for 500K steps with a batch size of 8K and a sequence length of 512. The optimizer used is Adam with a learning rate of 6e-4, $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 1e-6$, a weight decay of 0.01, learning rate warmup for 24,000 steps and linear decay of the learning rate after.

Evaluation results

When fine-tuned on downstream tasks, this model achieves the following results:

Glue test results:

Task	MNLI	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE
	87.6	91.9	92.8	94.8	63.6	91.2	90.2	78.7

BibTeX entry and citation info

```
@article{DBLP:journals/corr/abs-1907-11692,
  author    = {Yinhan Liu and
               Myle Ott and
               Naman Goyal and
```

```
                 Jingfei Du and
                 Mandar Joshi and
                 Danqi Chen and
                 Omer Levy and
                 Mike Lewis and
                 Luke Zettlemoyer and
                 Veselin Stoyanov},
  title     = {RoBERTa: {A} Robustly Optimized {BERT} Pretraining Appro
  journal   = {CoRR},
  volume    = {abs/1907.11692},
  year      = {2019},
  url       = {http://arxiv.org/abs/1907.11692},
  archivePrefix = {arXiv},
  eprint    = {1907.11692},
  timestamp = {Thu, 01 Aug 2019 08:59:33 +0200},
  biburl    = {https://dblp.org/rec/journals/corr/abs-1907-11692.bib},
  bibsource = {dblp computer science bibliography, https://dblp.org}
}
```

Visualize in exBERT lite

System theme

Company

TOS

Privacy

About

Jobs