قيقدت جذومن

```
import os
import glob
import json
               import load_dotenv
           import AutoTokenizer, AutoModelForCausalLM, Trainer, TrainingArguments, DataCollatorForLanguageMode
            import Dataset, load_dataset
import torch

load_dotenv()

      _ = "deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B"  #
       _ = "trained_model"
      _ = "train.jsonl"
    _    _ = 512
     _ = 8
     = 3

def       _  )    _:(     _
          _ = []
    for      _ in os.listdir(  :(     _
             _ = os.path.join(       _,    (   _
        if not os.path.isdir(  :(   _

        for      _ in glob.glob(os.path.join(     _, "*.md")):
                  _ _   _ =     ) _   _, 'r', encoding='utf-8')
                = ()read.
            #
                = )split.   "---", 2)[-1].strip()
            (   )append.
        except Exception as e:
            print(f"              }_{   : {e}")
    return

def       _  )    _, :(
        =   )    , truncation=True, padding=True, max_length=MAX_LENGTH, return_tensors="pt")
            from.    __dict(   )

def       )    _    _, :(
```

```
            _ = TrainingArguments(
        output_dir=OUTPUT_DIR,
        overwrite_output_dir=True,
        num_train_epochs=EPOCHS,
        per_device_train_batch_size=BATCH_SIZE,
        save_steps=10_000,
        save_total_limit=2,
        prediction_loss_only=True,
        remove_unused_columns=False,
    )
         = AutoModelForCausalLM.from_pretrained(MODEL_NAME, trust_remote_code=True)
          _ = DataCollatorForLanguageModeling(     =     , mlm=False)
        = Trainer(
        model=    ,
        args=          _,
        train_dataset=        _,
        data_collator=      _,
    )
    ()train.
    save.    _model(OUTPUT_DIR)

def  :()  _
           _ = "_posts"
        =       _  )    _(     _
          = LlamaTokenizerFast.from_pretrained(MODEL_NAME, trust_remote_code=True, use_fast=True)
    pad.      _token = eos.      _token
           _ =       _   )     _, (
         )    _     _, (

if __name__ == "__main__":
     ()  _
```