

به نام خدا

تکلیف سری هفتم NLP

بهاره کاوسی نژاد – 99431217

عملکرد هر مدل بسته به نوع و حجم دیتای ورودی متغیر است. در هر سه مدل در Zero-shot learning، مدل از قبل چیزی را آموزش ندیده است اما به تدریج با 1 و 5 داده آموزشی دارای عملکرد بهتری خواهد بود.

همچنین نوع prompt ورودی نیز بر پاسخ مدل تاثیر گذار است. باید prompt طوری انتخاب شود که مفهوم، داده ها (در دو قسمت One-shot learning و Five-shot learning) و خواسته مسئله با دقت به مدل داده شود.

همچنین به دلیل فارسی بودن دیتاست، مدل هایی که به صورت ویژه روی زبان فارسی تمرکز دارند بهتر عمل می کنند.

می توان با بدست آوردن امتیازهای متفاوت این عملکرد را به صورت دقیق تر مقایسه نمود.

در ادامه به توضیح کد می پردازیم:

پس از نصب و import کردن کتابخانه های لازم، به load کردن مدل می پردازیم.

```
q_config = BitsAndBytesConfig (
    load_in_4bit = True,
    bnb_4bit_quant_type="nf4",
)
```

این قسمت برای quantization است تا بتوانیم مدل های بزرگتر را load کنیم.

```
def create_model_tokenizer(name, q_config):
    tokenizer = AutoTokenizer.from_pretrained(name)
    model = AutoModelForCausalLM.from_pretrained (
        name,
        quantization_config = q_config,
        device_map="cuda",
        torch_dtype="auto",
        trust_remote_code = True
    )
    return model, tokenizer
```

از این تابع به منظور تعریف مدل و tokenizer استفاده می کنیم.

```
# Load model directly
model1, tokenizer1 = create_model_tokenizer("NousResearch/Meta-Llama-3-8B-Instruct", q_config)
```

مدل را می سازیم. در این جا به عنوان مثال مدل اول آورده شده است.

```
generation_args = {
    "max_new_tokens": 500,
    "return_full_text": False,
```

```

"temperature": 0.0,
"do_sample": False,
}

```

Generation_args را تعریف می کنیم.

```
dataset = load_dataset("persiannlp/parsinlu_entailment")
```

دیتاست مورد نظر را لود می کنیم.

```

questions = random.choice(dataset["train"])
prompt4 = f"Question 1: {questions['sent1']}\nQuestion 2:
{questions['sent2']}\nAre these questions similar? Just say 'yes' or 'no'"

```

Prompt را با استفاده از یک دیتای تصادفی از دیتاست تشکیل می دهیم و در ادامه آن را به مدل می دهیم و خروجی را چاپ می کنیم.

```

messages4 = [
    {"role": "user", "content": prompt4},
]
pipe2 = pipeline(
    "text-generation",
    model=model2,
    tokenizer=tokenizer2,
)
output4 = pipe2(messages4, **generation_args)
print(output4[0]['generated_text'])

```

تفاوت قسمت های One-shot Learning و Five-shot Learning با Zero-shot Learning در تشکیل prompt است:

One-shot Learning:

```

train2 = random.choice(dataset["train"])
test2 = random.choice(dataset["test"])
# question1 = "How old are you?"
# question2 = "What is your age?"
prompt5 = f"Sentence 1: {train2['sent1']}\nSentence 2: {train2['sent2']}\nThe
label for these 2 questions is:{train2['label']}\nNow tell me this: Are these
questions similar? Just say 'yes' or 'no'\n Sentence 3: {test2['sent1']}\n
Sentence 4: {test2['sent2']}"

```

Five-shot Learning:

```

prompt6 = (
    f"Sentence 1: {fiveshot_train2[0]['sent1']}\nSentence 2:
{fiveshot_train2[0]['sent2']}\nThe label for these 2 questions
is:{fiveshot_train2[0]['label']};\n"
    f"Sentence 3: {fiveshot_train2[1]['sent1']}\nSentence 4:
{fiveshot_train2[1]['sent2']}\nThe label for these 2 questions
is:{fiveshot_train2[1]['label']};\n"

```

```
        f"Sentence 5: {fiveshot_train2[2]['sent1']}\nSentence 6:  
{fiveshot_train2[2]['sent2']}\nThe label for these 2 questions  
is:{fiveshot_train2[2]['label']};\n"  
        f"Sentence 7: {fiveshot_train2[3]['sent1']}\nSentence 8:  
{fiveshot_train2[3]['sent2']}\nThe label for these 2 questions  
is:{fiveshot_train2[3]['label']};\n"  
        f"Sentence 9: {fiveshot_train2[4]['sent1']}\nSentence 10:  
{fiveshot_train2[4]['sent2']}\nThe label for these 2 questions  
is:{fiveshot_train2[4]['label']};\n"  
        f"Now tell me this: Are these questions similar? Just say 'yes' or  
'no'\n Sentence 3: {fiveshot_test2[0]['sent1']}\n Sentence 4:  
{fiveshot_test2[0]['sent2']}"  
    )
```