

ДОДАТОК А

Зразки згенерованих текстів

У додатку наведено приклади текстів, згенерованих базовою та донавченою моделями за однаковими початковими запитами (промптами). Приклади ілюструють відмінності у зв'язності, граматичній коректності та збереженні тематичного стилю.

Запит 1: «Президент країни під час засідання уряду заявив, що»

Базова модель GPT-Neo 125M (без донавчання):

«Президент країни під час засідання уряду заявив, що the government will continue до реформ, але точні terms поки не visnacheni. Він also додав, що питання буде розглянуто later цього року.»

Донавчена модель GPT-Neo 125M + LoRA:

«Президент країни під час засідання уряду заявив, що пріоритетом залишається підтримка економічної стабільності та реалізація запланованих реформ. За його словами, відповідні рішення буде ухвалено найближчим часом після консультацій із профільними міністерствами.»

Запит 2: «За даними метеорологів, найближчими днями»

Базова модель GPT-Neo 125M (без донавчання):

«За даними метеорологів, найближчими днями weather буде changeable, з temperatures близько 10 degrees. Можливі опади in деяких regions країни.»

Донавчена модель GPT-Neo 125M + LoRA:

«За даними метеорологів, найближчими днями в більшості областей країни очікується мінлива хмарність та невеликі опади. Температура повітря вночі становитиме від 5 до 10 градусів тепла, удень – до 15 градусів. Синоптики прогнозують поступове потепління наприкінці тижня.»

ДОДАТОК Б

Фрагменти програмного коду системи

У додатку наведено ключові фрагменти програмної реалізації системи генерації тексту: налаштування донавчання моделі методом LoRA, реалізацію циклу навчання та функцію генерації тексту.

Лістинг Б.1 – Налаштування донавчання моделі методом LoRA

```
from transformers import AutoModelForCausalLM, AutoTokenizer
from peft import LoraConfig, get_peft_model

model_name = "EleutherAI/gpt-neo-125M"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(model_name)

lora_config = LoraConfig(
    r=16,
    lora_alpha=32,
    target_modules=["q_proj", "v_proj"],
    lora_dropout=0.05,
    bias="none",
    task_type="CAUSAL_LM",
)

model = get_peft_model(model, lora_config)
model.print_trainable_parameters()
```

Лістинг Б.2 – Реалізація циклу навчання моделі

```
from transformers import Trainer, TrainingArguments
from transformers import DataCollatorForLanguageModeling

training_args = TrainingArguments(
    output_dir="./gpt-neo-uk-lora",
    num_train_epochs=3,
```

```

    per_device_train_batch_size=8,
    gradient_accumulation_steps=4,
    learning_rate=2e-4,
    warmup_steps=100,
    logging_steps=50,
    eval_strategy="epoch",
    save_strategy="epoch",
    fp16=True,
)

data_collator = DataCollatorForLanguageModeling(
    tokenizer=tokenizer, mlm=False)

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=val_dataset,
    data_collator=data_collator,
)

trainer.train()
model.save_pretrained("./gpt-neo-uk-lora")

```

Лістинг Б.3 – Функція генерації тексту

```

import torch

def generate_text(prompt, max_new_tokens=120,
                  temperature=0.8, top_p=0.9):
    inputs = tokenizer(prompt, return_tensors="pt")
    with torch.no_grad():
        output = model.generate(
            **inputs,
            max_new_tokens=max_new_tokens,
            do_sample=True,
            temperature=temperature,

```

```
        top_p=top_p,  
        repetition_penalty=1.2,  
        pad_token_id=tokenizer.eos_token_id,  
    )  
    return tokenizer.decode(output[0],  
                           skip_special_tokens=True)
```

ДОДАТОК В

Відомість кваліфікаційної роботи

Позначення					Найменування		Дод. відомості	
					Текстові документи			
1.					Пояснювальна записка		64 с.	