Конечно! Вот **полный, структурированный и подробно размеченный гайд** по обучению **каждого из агентов** (Mixtral-8x7B), включая:

* Формат и примеры сложной разметки данных.
* Полный пайплайн дообучения.
* Конкретные сниппеты кода.

**🚀 Полный гайд по дообучению модели Mixtral-8x7B**

**Стек технологий:**

* **Python**, Hugging Face Transformers, PEFT (LoRA), pdfplumber, camelot.
* GPU: NVIDIA A100 (40+GB VRAM) или аналогичный.

**🟢 Агент №1: Извлечение структурированных данных из документов**

**Формат разметки данных для обучения**

{

"instruction": "Из документа извлеки выручку, EBITDA и чистую прибыль компании.",

"input": "<полный текст финансового отчета>",

"output": {

"Выручка": "340.1 млрд руб.",

"EBITDA": "48.7 млрд руб.",

"Чистая прибыль": "15.5 млрд руб."

}

}

**⚠️ Сложный пример (с дополнительными параметрами):**

{

"instruction": "Из отчета извлеки выручку, EBITDA, чистую прибыль, GMV электронной коммерции, количество активных пользователей и рентабельность EBITDA.",

"input": "<полный текст финансового отчета>",

"output": {

"Выручка": "1 094.6 млрд руб.",

"EBITDA": "188.6 млрд руб.",

"Чистая прибыль": "11.5 млрд руб.",

"GMV электронной коммерции": "956.3 млрд руб.",

"Активные пользователи": "53.2 млн чел.",

"Рентабельность EBITDA": "17.2%"

}

}

**Сниппет дообучения Mixtral (LoRA fine-tuning):**

from datasets import load\_dataset

from transformers import AutoModelForCausalLM, AutoTokenizer, BitsAndBytesConfig

from peft import LoraConfig, get\_peft\_model, prepare\_model\_for\_kbit\_training

import torch

from transformers import Trainer, TrainingArguments

model\_id = "mistralai/Mixtral-8x7B-Instruct-v0.1"

bnb\_config = BitsAndBytesConfig(

load\_in\_4bit=True,

bnb\_4bit\_use\_double\_quant=True,

bnb\_4bit\_quant\_type="nf4",

bnb\_4bit\_compute\_dtype=torch.bfloat16,

)

tokenizer = AutoTokenizer.from\_pretrained(model\_id)

model = AutoModelForCausalLM.from\_pretrained(model\_id, quantization\_config=bnb\_config, device\_map='auto')

model = prepare\_model\_for\_kbit\_training(model)

peft\_config = LoraConfig(r=32, lora\_alpha=64, target\_modules=["q\_proj", "v\_proj"], task\_type="CAUSAL\_LM")

model = get\_peft\_model(model, peft\_config)

data = load\_dataset('json', data\_files='train\_agent1.jsonl')

def format\_instruction(example):

instr = example['instruction']

input\_data = example['input']

output\_data = example['output']

return f"Задача: {instr}\nДокумент: {input\_data}\nОтвет: {output\_data}"

data = data.map(lambda x: tokenizer(format\_instruction(x), truncation=True, padding="max\_length", max\_length=2048), batched=True)

training\_args = TrainingArguments(

output\_dir="./results\_agent1",

per\_device\_train\_batch\_size=2,

gradient\_accumulation\_steps=4,

num\_train\_epochs=3,

learning\_rate=2e-4,

fp16=True

)

trainer = Trainer(model=model, args=training\_args, train\_dataset=data["train"])

trainer.train()

model.save\_pretrained("mixtral\_agent1")

tokenizer.save\_pretrained("mixtral\_agent1")

**🟢 Агент №2: Подготовка данных для имитационного моделирования**

**Формат разметки данных:**

{

"instruction": "Из документа извлеки численность клиентов, средний чек и процент оттока для создания имитационной модели.",

"input": "<полный текст отчета>",

"output": {

"Клиенты": "53.2 млн",

"Средний чек": "1200 руб.",

"Отток": "4.5%"

}

}

**⚠️ Сложный пример (более детально):**

{

"instruction": "Извлеки данные для имитации рынка: количество клиентов, средний чек, отток, расходы на маркетинг, долю рынка и выручку.",

"input": "<полный текст отчета>",

"output": {

"Клиенты": "53.2 млн",

"Средний чек": "1200 руб.",

"Отток": "4.5%",

"Расходы на маркетинг": "15 млрд руб.",

"Доля рынка": "66.4%",

"Выручка": "340.1 млрд руб."

}

}

**Код обучения аналогичен агенту №1, но с другими данными (train\_agent2.jsonl).**

**🟢 Агент №3: Аналитика и рекомендации по результатам моделирования**

**Формат разметки данных с CoT:**

{

"instruction": "Проанализируй результаты имитации и дай рекомендации по улучшению прибыли компании.",

"input": "Выручка снизилась на 5%. Расходы на маркетинг неизменны. Отток клиентов увеличился на 2%.",

"output": {

"reasoning": "Рост оттока вызвал снижение выручки. Недостаточная эффективность текущих маркетинговых стратегий не позволяет удерживать клиентов.",

"recommendations": [

"Увеличить бюджет на маркетинг на 10%",

"Ввести программу лояльности для удержания клиентов"

]

}

}

**⚠️ Сложный пример с глубокой аналитикой:**

{

"instruction": "Оцени результаты агентной модели и предоставь подробные рекомендации для повышения EBITDA и снижения оттока.",

"input": "EBITDA снизилась на 7%, отток вырос до 6%, маркетинговый бюджет стабилен.",

"output": {

"reasoning": "EBITDA снизилась на фоне роста оттока. Стабильный бюджет на маркетинг не позволяет компенсировать снижение лояльности клиентов. Нужно срочно повышать эффективность маркетинга и удержания клиентов.",

"recommendations": [

"Повысить расходы на таргетированную рекламу на 15%",

"Реализовать скидочные программы для удержания клиентов с высоким риском оттока"

],

"expected\_impact": {

"EBITDA": "+4%",

"Отток": "-2.5%"

}

}

}

**Скрипт дообучения идентичен агенту №1, но с изменённой функцией форматирования инструкций:**

def format\_instruction\_agent3(example):

instr = example['instruction']

input\_data = example['input']

output = example['output']

reasoning = output['reasoning']

recommendations = ", ".join(output['recommendations'])

return f"Задача: {instr}\nДанные: {input\_data}\nАнализ: {reasoning}\nРекомендации: {recommendations}"

Используй файл данных (train\_agent3.jsonl).