Kyrgyz text completion

Build a Kyrgyz text generator using the mGPT model to auto-complete prompts
(like weather/news) and process raw data for AI training.

```python
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM
import pandas as pd
from datasets import Dataset
```

```python
class TextGenerator:
    def __init__(self, model_name="ai-forever/mGPT-1.3B-kirgiz"):
        self.tokenizer = AutoTokenizer.from_pretrained(model_name)
        self.model = AutoModelForCausalLM.from_pretrained(model_name)
        self.device = "cuda" if torch.cuda.is_available() else "cpu"
        self.model.to(self.device)

    def generate(self, prompt, max_length=50, temperature=0.9):
        inputs = self.tokenizer(prompt, return_tensors="pt").to(self.device)
        outputs = self.model.generate(
            **inputs,
            max_new_tokens=max_length,
            temperature=temperature,
            do_sample=True
        )
        return self.tokenizer.decode(outputs[0], skip_special_tokens=True)
```

```python
    def fine_tune(self, dataset, epochs=3, save_steps=500):
        def tokenize_function(examples):
            return self.tokenizer(examples["prompt"], examples["completion"],
                                  truncation=True, padding="max_length",
                                  max_length=128, return_tensors="pt")

        tokenized_dataset = dataset.map(tokenize_function, batched=True)

        training_args = TrainingArguments(
            output_dir="./fine_tuned_model",
            overwrite_output_dir=True,
            num_train_epochs=epochs,
            per_device_train_batch_size=4,
            save_steps=save_steps,
            save_total_limit=2,
            logging_dir="./logs",
            logging_steps=100,
            learning_rate=5e-5,
            warmup_steps=500,
            weight_decay=0.01,
            fp16=torch.cuda.is_available(),
        )

        trainer = Trainer(
            model=self.model,
            args=training_args,
            train_dataset=tokenized_dataset,
        )

        trainer.train()

        trainer.save_model("./fine_tuned_model")
```

```python
class DataPreprocessor:
    @staticmethod
    def load_and_split_data(file_path, num_words=8):
        df = pd.read_csv(file_path, header=None, names=["full_text"])
        def split_text(text):
            words = str(text).split()
            prompt = " ".join(words[:num_words])
            completion = " ".join(words[num_words:]) if len(words) > num_words else ""
            return pd.Series([prompt, completion])

        df[["prompt", "completion"]] = df["full_text"].apply(split_text)
        return Dataset.from_pandas(df[["prompt", "completion"]])
```

```
preprocessor = DataPreprocessor()
generator = TextGenerator()
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: Use
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access pu
  warnings.warn(
```

```
dataset = preprocessor.load_and_split_data("all_texts.txt")
```

```
print("Начинаем дообучение модели...")
generator.fine_tune(dataset, epochs=3, save_steps=500)
print("Дообучение завершено!")
```

```
prompt = "Бүгүнкү аба ырайы"
output = generator.generate(prompt, max_length=500)
print("\nOutput:", output)
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.

Output: Бүгүнкү аба ырайынын кескин өзгөрүшү кескин өсүп, андагы балдардын дагы эсеби өсүп жатат.
```

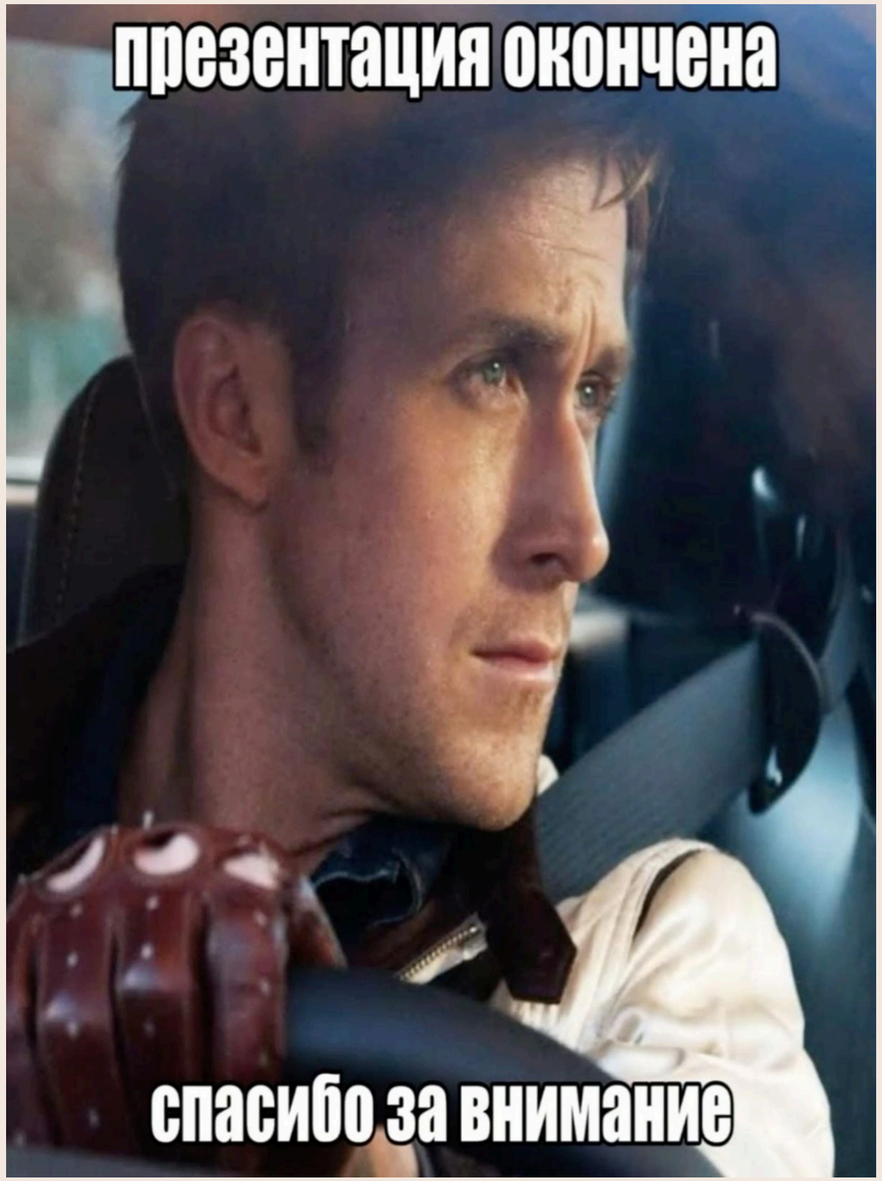
Output: Бүгүнкү аба ырайынын кескин өзгөрүшү кескин өсүп, андагы балдардын дагы эсеби өсүп жатат. Бүгүнкү күнү Бишкектеги жер титирөө катталганы маалым болду. Учурда Жалал-Абад шаары боюнча 10 балдар арасында жер титирөө болду. ӨКМ: Кыргызстанда коронавирус жуктургандардын саны 668ге жетти - "Кабар" - ДЕМ Өлкөдө жугуштуу оорулардын алдын алуу программасы бир катар эл аралык уюмдардан, Жогорку Кеңештин, өкмөттүн

```
prompt = "Бүгүнкү аба ырайы"
output = generator.generate(prompt, max_length=64)
print("\nСгенерированный текст после дообучения:")
print(output)
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.

Сгенерированный текст после дообучения:
Бүгүнкү аба ырайынын кооптуулугунан чыккан кыргыз жери кыйрап кетти. Бул шейшемби күнү айрым жарандар кыйнашты.
Батыштан соккон шамалдын
```

```
generator.save_model("./final_model")
print("Модель сохранена в папке 'final_model'")
```


презентация окончена
спасибо за внимание


SIGMA BOY
LEO BEATS