УДК 004

Разработка программы для генерации текста на основе GPT

Андриенко Иван Сергеевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье рассматривается процесс разработки программы для генерации текста на основе GPT (Generative Pre-trained Transformer). Описаны основные этапы создания программы, включая интеграцию GPT, настройку генерации текста и обработку результатов. Программа предназначена для автоматического создания текстов с учетом заданных параметров и использует библиотеку Hugging Face для доступа к модели GPT. Приведены примеры использования программы для генерации текстов различной тематики. Показано, как GPT может быть использован для автоматизации процесса создания текстов и получения осмысленных и связных результатов.
Ключевые слова: GPT, генерация текста, Hugging Face, автоматизация, машинное обучение.

Development of a program for generating text based on GPT

Andrienko Ivan Sergeevich
Sholom-Aleichem Priamursky State University
Student

Abstract
This article discusses the process of developing a program for generating text based on GPT (Generative Pre-trained Transformer). The main stages of creating the program are described, including GPT integration, setting up text generation and processing results. The program is designed to automatically create texts based on the specified parameters and uses the Hugging Face library to access the GPT model. Examples of using the program to generate texts on various topics are given. It is shown how GPT can be used to automate the process of creating texts and obtaining meaningful and coherent results.
Keywords: GPT, text generation, Hugging Face, automation, machine learning.

1 Введение

1.1 Актуальность

Автоматизация процесса создания текста с использованием моделей машинного обучения становится все более востребованной в условиях цифровизации и растущего объема информации. Одной из наиболее мощных моделей для генерации текста является GPT, разработанная компанией

OpenAI. Эта модель позволяет генерировать связные и осмысленные тексты на основе заданных вводных данных, что делает ее полезным инструментом в различных областях, включая журналистику, маркетинг, научные исследования и программирование. Использование GPT в автоматизированных системах позволяет значительно сократить временные затраты на создание текстов, а также улучшить их качество и разнообразие.

1.2 Обзор исследований

В своей работе А. Куанушбек исследовал применение нейронных сетей для автоматической генерации текстов в своем исследовании, где описал использование различных моделей машинного обучения для создания текстов на основе заданных параметров [1]. В работе С.А. Проскурина и Л.В. Гаев исследуется влияние различных параметров обучения модели GPT на качество генерируемого текста, авторы предлагают методы оптимизации и улучшения производительности модели [2]. П.А. Сергеев в статье рассматривает особенности использования GPT-3 для автоматической генерации текстов и анализирует его возможности в контексте написания статей и отчетов [3]. А.И. Власов, Д.Н. Крамарь, А.А. Цветков в исследовании анализируют методы настройки GPT для генерации текстов различной сложности и структуры, а также исследуют вопросы оптимизации модели для конкретных задач [4]. О.А. Смелов проводит сравнительный анализ различных моделей генерации текста и их применимость в разных сценариях, включая GPT, T5 и другие [5].

1.3 Цель исследования

Цель исследования – разработать программу для генерации текста на основе модели GPT, обеспечивающую автоматизацию процесса создания текстов различной тематики. В рамках исследования будут рассмотрены основные этапы интеграции GPT, настройки параметров генерации текста и обработки результатов для достижения наилучших показателей качества.

2 Материалы и методы

Для разработки программы использовалась библиотека Hugging Face Transformers, предоставляющая доступ к модели GPT. Программа была написана на языке программирования Python. В качестве среды разработки использовался сервис google colab.

3 Результаты и обсуждения

GPT – мощная модель генерации текста, основанная на трансформерах, которая способна создавать связные и осмысленные тексты на основе заданных параметров.

В процессе разработки и тестирования программы для генерации текста на основе GPT будет использован Google Colab. Эта облачная платформа предоставляет удобную среду для работы с Python и специализированными библиотеками для машинного обучения. Google Colab

позволяет эффективно выполнять код, управлять ресурсами и настраивать модель GPT, предоставляя доступ к мощным вычислительным ресурсам, включая GPU и TPU, что существенно ускоряет процесс обучения и генерации текстов.

Для работы с моделью GPT-2 используются две основные библиотеки: torch и transformers. Библиотека torch предоставляет инструменты для работы с нейронными сетями, а transformers включает в себя готовые модели и токенизаторы. В данном коде используется модель GPT-2, которую загружают с помощью GPT2LMHeadModel и GPT2Tokenizer, позволяя преобразовать текст в формат, понятный модели, и генерировать текст на основе входных данных (рис. 1).

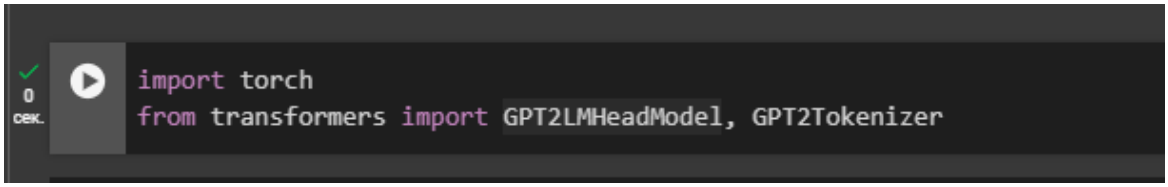

Рис. 1. Подключение библиотек

Далее происходит загрузка предварительно обученной модели GPT-2 и соответствующего токенайзера. Сначала задается название модели "gpt2", после чего с помощью GPT2LMHeadModel.from_pretrained загружается сама модель. Токенайзер, который преобразует текст в числовые последовательности для обработки моделью, загружается аналогичным образом через GPT2Tokenizer.from_pretrained (рис. 2).

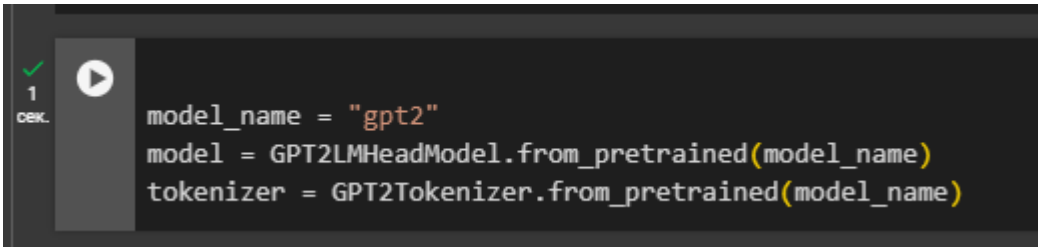

Рис. 2. Загрузка модели

Создадим функцию generate_text, которая позволяет генерировать текст на основе заданного начального фрагмента (prompt). Функция принимает несколько параметров, включая максимальную длину текста (max_length), температуру для регулирования креативности генерации (temperature), и количество возвращаемых последовательностей (num_return_sequences).

Внутри функции сначала происходит преобразование начального текста в числовую последовательность с помощью токенайзера (tokenizer.encode). Затем модель генерирует текст, используя эту последовательность в качестве входных данных. В процессе генерации учитываются различные параметры, такие как максимальная длина текста и температура (рис. 3).

```python
def generate_text(prompt, max_length=100, temperature=1.0, num_return_sequences=1):

    inputs = tokenizer.encode(prompt, return_tensors="pt")
    outputs = model.generate(
        inputs,
        max_length=max_length,
        temperature=temperature,
        num_return_sequences=num_return_sequences,
        no_repeat_ngram_size=2,
        pad_token_id=tokenizer.eos_token_id
    )

    generated_text = tokenizer.decode(outputs[0], skip_special_tokens=True)

    return generated_text
```

Рис. 3. Создание функции generate_text

Добавим функцию для генерации несколько вариантов текста на основе одного и того же начального промта. Функция принимает два параметра: начальный текст и количество вариантов, которые необходимо создать.

Внутри функции создается пустой список texts, куда будут добавляться сгенерированные варианты. Затем в цикле вызывается функция generate_text для генерации каждого варианта с параметром temperature=0.9, что влияет на разнообразие генерируемого текста. Сгенерированные варианты добавляются в список, который затем возвращается в виде результата (рис. 4).

```python
def generate_multiple_variants(prompt, num_variants=3):
    texts = []
    for _ in range(num_variants):
        text = generate_text(prompt, temperature=0.9)
        texts.append(text)
    return texts
```

Рис. 4. Создание функции generate_multiple_variants

Для контроля вывода рекомендуется проверять длину текста. Функция может быть полезна для анализа текста от gpt, например, чтобы убедиться, что сгенерированный текст соответствует необходимым ограничениям по длине (рис. 5).

```python
def calculate_text_length(text):
    return len(text.split())
```

Рис. 5. Создание функции для оценки длины сгенерированного текста

Зададим основные параметры для генерации текста. В качестве промта используется текст "tell me about yourself". Затем задаются параметры максимальной длины текста (max_length), температуры (temperature) и количества генерируемых вариантов (num_return_sequences) (рис. 6). В итоге сгенерированный текст выводится на экран (рис. 7).

Рис. 6. Настройка параметров для генерации текста

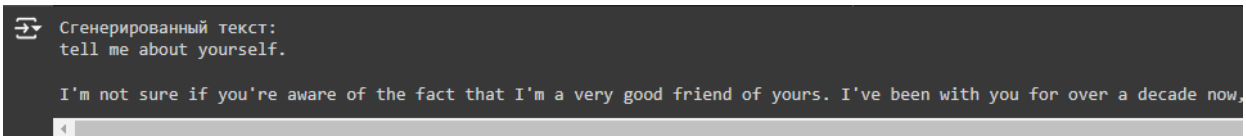

Рис. 7. Сгенерированный текст

Выводы

В данной статье представлен процесс разработки программы для генерации текста на основе GPT. Рассмотрены ключевые аспекты интеграции модели GPT, настройки параметров генерации текста и использования дополнительных инструментов, для улучшения качества результатов. Программа демонстрирует способность GPT создавать связные и осмысленные тексты, что делает ее полезным инструментом для автоматизации процесса написания текстов. Разработка может быть использована в различных сферах, включая журналистику, маркетинг и научные исследования.

Библиографический список

1. Любимов И.А., Бугаков П.Ю. Разработка Telegram бота для бесплатного использования ChatGPT в России // Интерэкспо Гео-Сибирь. 2023. Т. 7. № 2. С. 51-55.
2. Jin J., Kim M. GPT-Empowered Personalized eLearning System for Programming Languages // Applied Sciences (Switzerland). 2023. Т. 13. № 23. С. 12773.
3. Минаев О.М., Разваров А.В. Интеграция Chat GPT с информационной системой учета продаж на основе микросервиса FastAPI // В сборнике: Мухтаровские чтения: Актуальные проблемы математики, методики ее преподавания и смежные вопросы. Материалы международной научной

конференции. Махачкала, 2024. С. 113-117.
4. Дубровский В.В. Применение платформы машинного обучения Hugging Face для разработки интеллектуальных систем автоматического реферирования текстовых документов // В книге: Актуальные проблемы современной науки, техники и образования. Тезисы докладов 82-й международной научно-технической конференции. Магнитогорск, 2024. С. 225.
5. Береснев Д.В. Анализ подходов к использованию предобученных моделей в разработке корпоративных чат-ботов // В сборнике: Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях. Сборник статей научно-технического семинара студентов кафедры «Инженерной кибернетики». Москва, 2023. С. 68-77.