Обучение LLM для заказа билета

Галяутдинов Акар

Ноябрь 2023

https://github.com/7Askar7/LLaMa_Train

Аннотация

Данный проект нацелен на то, чтобы LLM помогала пользователю заказать билет на самолет. Для этого будет собран и размечен датасет, и обучена модель LLaMa-2-7b.

1 Введение

Мы сталкиваемся с множеством приложений и веб-сервисов для заказа билетов на самолет, но пользователь тратит свое время на рутинные задачи, которые можно автоматизировать. В настоящее время уже есть функции, которые сохраняют данные пользователя, но они используют часто используемую информацию. Наше решение нацелено на то, чтобы пользователь ввел только информацию поездке и данные из его запроса были определены и на основе них можно заказать билет.

1.1 Команда

Данный проект был создан Галяутдиновым Аскаром.

Галяутдинов Аскар – собрал датасет, подготовил данные для обучения, обучил LLaMa-2.

2 Работа

История развития больших языковых моделей (LLM) насчитывает уже несколько десятилетий. Первые LLM были разработаны в 1950-х годах, но они были очень маленькими и не могли выполнять сложные задачи.

В 1980-х годах появились более крупные LLM, но они все еще были относительно медленными и неэффективными. В 1990-х годах появились первые LLM, которые могли выполнять генерацию текста, перевод языков и ответы на вопросы.

В 2000-х годах произошел значительный прогресс в развитии LLM. Были разработаны новые методы обучения LLM, которые позволили создавать более крупные и сложные модели. В 2010-х годах появились LLM, которые могли выполнять такие задачи, как создание различных творческих текстовых форматов, таких как стихи, код, сценарии, музыкальные произведения, электронные письма, письма и т. д.

В настоящее время LLM являются одними из самых передовых технологий в области искусственного интеллекта. Они используются в различных приложениях, включая машинный перевод, ответы на вопросы, создание контента и обучение.

3 Данные

Чтобы обучить LLaMa-2 так, чтобы она могла заказать билет, нам необходимы данные. Изучив популярные сайты в сфере искусственного интеллекта, такие как HuggingFace, Kaggle, GitHub, мне удалось найти необходимые датасеты только на английском языке, но целевой язык – русский.

Было принято попробовать применить популярный метод – дистилляция. ChatGPT-3.5 справлялся с этим не всегда хорошо, так как

многие данные повторялись или выходили за рамки поставленной задачи. Тогда я решил передавать данные из ChatGPT-3.5 в Bard. ChatGPT-3.5 занимался генерацией предложений, а Bard извлекал сущности и переносил их в Excel-таблицу, но без исправлений со стороны человека не обошлось.

После сбора данных необходимо было сконвертировать их в формат CSV, где каждая строка соответствует строке таблицы, а значения внутри строки разделены запятыми. CSV-файлы легко обрабатываются в Python. После предобработки и конвертации данных в нужный формат, мы разделили их на test_data и train_data, где размеры test_data = 8 примеров, а train_data = 71.

4 Модель

Используем модель LLaMa-2-7b так как на сегодняшний день она является одной из лучшей open-source LLM.

4.1 Llama-2-7b

Архитектура LLama2-7b основана на архитектуре Transformer, которая является одной из наиболее эффективных архитектур для LLM. Transformer использует механизм внимания для понимания взаимосвязей между словами в предложении.

Новый механизм внимания, используемый в LLama2-7b, называется "Multi-head attention". Multi-head attention использует несколько параллельных механизмов внимания, которые работают на разных уровнях абстракции. Это позволяет LLama2-7b лучше понимать взаимосвязи между словами в предложении, даже если они находятся далеко друг от друга.

Новый слой преобразования, используемый в LLama2-7b, называется "Transformer decoder". Transformer decoder использует механизм внимания для генерации нового текста. Он также использует слой преобразования, который позволяет модели генерировать различные творческие текстовые форматы.

4.1.1 Механизм внимания

Механизм внимания - это ключевой компонент архитектуры Transformer. Он позволяет модели понимать взаимосвязи между словами в предложении. Механизм внимания работает следующим образом:

Модель сначала вычисляет матрицу внимания, которая представляет собой оценку важности каждого слова в предложении. Затем модель использует матрицу внимания для вычисления выходного значения для каждого слова.

4.1.2 Multi-head attention

Multi-head attention использует несколько параллельных механизмов внимания, которые работают на разных уровнях абстракции. Это позволяет LLama2-7b лучше понимать взаимосвязи между словами в предложении, даже если они находятся далеко друг от друга.

4.1.3 Transformer decoder

Transformer decoder использует механизм внимания для генерации нового текста. Он также использует слой преобразования, который позволяет модели генерировать различные творческие текстовые форматы.

5 Метрики

Метрики для подсчета качества обучения модели используем LOSS , потому что это наиболее точный способ измерения того, насколько хорошо модель предсказывает фактические результаты.

$$\text{CrossEntropyLoss}(\hat{y}, y) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{T} y_{i,j}\log(\widehat{y_{i,j}})$$