Подсказки для 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 настроены подсказки.

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. Но обучать LLM - это очень дорогостоящий подход, ведь нам необходимы большие мощности для того, чтобы ее обучить. Конечно, стандартные компьютеры тоже могут подойти, но время ее обучения может составлять недели, а то и месяцы.

1.1 Команда

Данный проект был создан Галяутдиновым Аскаром.

Галяутдинов Аскар – собрал датасет, подготовил данные для обучения, настроил подсказки для Saiga.

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 Решение

4.1 DST Prompter

Учитывая диалог подсказчик DST направлен на то, чтобы помочь LLM предсказать состояние убеждений, используя инструкцию убеждения BI (Belief instructions).

Для каждой задачи/домена инструкция убеждения содержит название задачи/домена, все потенциальные имена слотов и их возможные значения.имена всех потенциальных слотов и их возможные значения . Что касается категориальных слотов, включены все возможные значения; в то время как для некатегориальных слотов, вводится только несколько примеров значений, например вводится только несколько примеров значений.

4.2 FAISS (Facebook AI Similarity Search)

FAISS - это библиотека для эффективного поиска похожих векторов в больших наборах данных. В контексте ScalableSemanticSearch FAISS используется для построения индекса и ускорения поиска:

Индексация векторов: FAISS позволяет строить индексы для эффективного хранения и поиска векторных представлений. В данном случае используется IndexFlatL2 (полный перебор) или IndexIVFPQ (индекс с кластеризацией и квантованием кодов) в зависимости от размера данных.

Оптимизированный поиск: После построения индекса FAISS предоставляет методы для выполнения быстрого и эффективного поиска похожих векторов. В данном случае используется поиск на основе косинусного расстояния.

Объединение ScalableSemanticSearch и FAISS позволяет создавать эффективные системы поиска, основанные на семантических векторах, что часто встречается в задачах обработки естественного языка и информационного поиска.

4.3 SAIGA2

Saiga2 - большая языковая модель, разработанная командой исследователей из Google AI. Она была обучена на огромном наборе данных из текстов и кода, и способна выполнять различные задачи.

Saiga2 использует архитектуру Transformer, которая является одним из самых передовых методов обработки естественного языка. Transformer состоит из нескольких слоев, которые обрабатывают текст последовательно, передавая информацию друг другу. Это позволяет модели лучше понимать контекст текста и выполнять сложные задачи.

Конкретно, Saiga2 использует архитектуру Transformer-XL, которая является расширенной версией Transformer. Transformer-XL позволяет модели лучше обрабатывать длинные последовательности текста, что делает ее более подходящей для задач, таких как перевод и вопросно-ответный режим.

Квантизация: Квантизация - это процесс преобразования чисел в более компактное представление. В случае Saiga2 веса модели преобразуются в представление с меньшей разрядностью. Это позволяет уменьшить размер модели без потери производительности.