LLM Tools & Structured Generation

Введение

Разработка LMки и приложений вокруг нее – трудоемкая задача. Хорошему разработчику для решения трудоемких задач требуются хорошие и удобные инструменты – об этом пойдет речь на нашем занятии

Карта разработки | Слайд 3

Работу с LLM можно разложить на несколько частей:

1. Нам нужны качественные данные, для этого нужно уметь их искать, размечать, очищать и иногда генерировать.
2. Нужны эффективные инструменты для обучения. Как для претрейна, так и для алайнмента. Стандартные инструменты вроде torch'a и tensorflow могут оказаться слишком низкоуровневыми, т.к. LMки обычно очень большие и необходимо решить много нюансов вокруг этого свойства.
3. Важно понимать, насколько хорошо модель работает на наших задачах. Для этого нужно выбрать бенчмарки и создать свои собственные. Эта потребность становится супер важной во время решения прикладных задач.
4. После того, как мы обучили наши модели нужно научиться их очень быстро и эффективно инференсить. LMки большие, нужно как-то с этим мириться. Обычный инференс через huggingface не эффективен.
5. Если вы хотите собрать сложную систему на основе LLM, может появиться потреность в чейнах и всяких библиотек для него.
6. Если вы хотите показать кому-то свое решение, то нужно подумать о том, в каком виде поднимать для него UI; Иными словами – как пользователь будет взаимодействовать с моделью?

Разметка и сбор фидбека | Слайды 4-7

Label Studio – для разметки данных. Можно запускать асессорские разметки любой сложности, настраивать валидацию, следить за согласованностью.
Argilla – среда для разметки данных вокруг LLM задач. Гораздо более приятный интерфейс и множество готовых решений для сбора LLM фидбека (рекомендую использовать ее)
distilabel – фреймворк для генерации синтетических данных. Генерируем SFT датасеты, преференсы и занимаемся их очисткой. Быстро, удобно, эффективно.

Обучение моделей | Слайд 8

Для претрейна и алайнмента LLM я рекомендую вам использовать готовые решения. Безусловно, вы можете попробовать самостоятельно реализовать все на вашем любимом фреймворке, однако в случае LLM возникает много вопросов, которые сложно решить с ходу.
Например, нужно писать грамотные датасеты и даталоадеры (объем данных огромен, и невозможно положить весь датасет в память); важно очень точно реализовать любые функциональности, связанные с параллелизмом; грамотно сохранять логи и чекпоинты.
Тут – небольшой список наиболее популярных фреймворков, которые могут вам пригодиться.