Введение

Как вы видели в Главе 1, модели трансформеров обычно очень большие. С миллионами и десятками *миллиардов* параметров, обучение и развертывание этих моделей - сложная задача. Кроме того, поскольку новые модели выходят практически ежедневно и каждая из них имеет свою собственную реализацию, попробовать их все - задача не из легких.

Для решения этой проблемы была создана библиотека 🤗 Transformers. Ее цель - предоставить единый API, с помощью которого можно загрузить, обучить и сохранить любую модель Transformer. Основными особенностями библиотеки являются:

- **Простота использования**: Скачать, загрузить и использовать современную модель NLP для инференса можно всего в две строчки кода.
- **Гибкость**: По своей сути все модели представляют собой простые классы PyTorch `nn.Module` или TensorFlow `tf.keras.Model` и могут быть обработаны как любые другие модели в соответствующих фреймворках машинного обучения (ML).
- **Простота**: В библиотеке почти нет абстракций. Концепция “Все в одном файле” является основной: прямой проход модели полностью определяется в одном файле, так что сам код понятен и доступен для изменения.

Эта последняя особенность делает 🤗 Transformers совершенно непохожей на другие ML-библиотеки. Модели не строятся на основе модулей которые совместно используются в разных файлах; вместо этого каждая модель имеет свои собственные слои. Помимо того, что это делает модели более доступными и понятными, это позволяет легко экспериментировать с одной моделью, не затрагивая другие.

Эта глава начнется со сквозного примера, в котором мы используем модель и токенизатор вместе, чтобы воссоздать функцию `pipeline()`, представленную в Главе 1. Далее мы обсудим API модели: мы погрузимся в модель и классы конфигурации, покажем, как загрузить модель и как она обрабатывает числовые данные для вывода прогнозов.

Затем мы рассмотрим API токенизатора, который является другим основным компонентом функции `pipeline()`. Токенизаторы берут на себя первый и последний шаги препроцессинга, обработку преобразования текста в числовые входы для нейронной сети и обратное преобразование в текст, когда это необходимо. Наконец, мы покажем вам, как обрабатывать несколько предложений, передавая их в модель в подготовленном батче, затем завершим все это более подробным рассмотрением высокоуровневой функции `tokenizer()`.

> ⚠️ Чтобы воспользоваться всеми возможностями, доступными в Model Hub и 🤗 Transformers, мы рекомендуем создать учетную запись.