Введение

В Главе 3 мы рассмотрели, как дообучить модель для конкретной задачи. При этом мы используем тот же токенизатор, на котором была предварительно обучена модель, но что делать, когда мы хотим обучить модель с нуля? В таких случаях использование токенизатора, который был предварительно обучен на корпусе из другой области или языка, как правило, является неоптимальным. Например, токенизатор, обученный на корпусе английских текстов, будет плохо работать на корпусе японских текстов, поскольку использование пробелов и знаков препинания в этих двух языках сильно отличается.

В этой главе вы узнаете, как обучить совершенно новый токенизатор на корпусе текстов, чтобы затем использовать его для предварительного обучения языковой модели. Все это будет сделано с помощью библиотеки 🤗 Tokenizers, которая предоставляет “быстрые” токенизаторы в библиотеке 🤗 Transformers. Мы подробно рассмотрим возможности, которые предоставляет эта библиотека, и выясним, чем быстрые токенизаторы отличаются от “медленных” версий.

Мы рассмотрим следующие темы:

- Как обучить новый токенизатор, аналогичный тому, который используется в данной контрольной точке, на новом корпусе текстов
- Особенности быстрых токенизаторов
- Различия между тремя основными алгоритмами токенизации по подсловам, используемыми в NLP сегодня
- Как создать токенизатор с нуля с помощью библиотеки 🤗 Tokenizers и обучить его на некоторых данных

Техники, представленные в этой главе, подготовят вас к разделу в Главе 7, где мы рассмотрим создание языковой модели по исходному коду Python. Для начала давайте разберемся, что значит “обучить” токенизатор.