Process text data

This guide shows specific methods for processing text datasets. Learn how to:

- Tokenize a dataset with map().
- Align dataset labels with label ids for NLI datasets.

For a guide on how to process any type of dataset, take a look at the general process guide.

Map

The map() function supports processing batches of examples at once which speeds up tokenization.

Load a tokenizer from Transformers:

```
>>> from transformers import AutoTokenizer

>>> tokenizer = AutoTokenizer.from_pretrained("bert-base-cased")
```

Set the `batched` parameter to `True` in the map() function to apply the tokenizer to batches of examples:

```
>>> dataset = dataset.map(lambda examples: tokenizer(examples["text"]), batched=True)
>>> dataset[0]
{'text': 'the rock is destined to be the 21st century\'s new " conan " and that he\'s going to mak
 'label': 1,
 'input_ids': [101, 1996, 2600, 2003, 16036, 2000, 2022, 1996, 7398, 2301, 1005, 1055, 2047, 1000,
 'token_type_ids': [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 'attention_mask': [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

The map() function converts the returned values to a PyArrow-supported format. But explicitly returning the tensors as NumPy arrays is faster because it is a natively supported PyArrow format. Set `return_tensors="np"` when you tokenize your text:

```
>>> dataset = dataset.map(lambda examples: tokenizer(examples["text"], return_tensors="np"), batch
```

Align

The align_labels_with_mapping() function aligns a dataset label id with the label name. Not all 🤗 Transformers models follow the prescribed label mapping of the original dataset, especially for NLI datasets. For example, the MNLI dataset uses the following label mapping:

```
>>> label2id = {"entailment": 0, "neutral": 1, "contradiction": 2}
```

To align the dataset label mapping with the mapping used by a model, create a dictionary of the label name and id to align on:

```
>>> label2id = {"contradiction": 0, "neutral": 1, "entailment": 2}
```

Pass the dictionary of the label mappings to the align_labels_with_mapping() function, and the column to align on:

```
>>> from datasets import load_dataset

>>> mnli = load_dataset("nyu-mll/glue", "mnli", split="train")
>>> mnli_aligned = mnli.align_labels_with_mapping(label2id, "label")
```

You can also use this function to assign a custom mapping of labels to ids.