Laporan Chapter 5 - NLP dengan Hugging Face

The Dataset Library

Nama Rey Rizqi Anugerah
Kelas TK45 01
NIM: 1103210146

Tujuan Chapter

1. Memahami peran dataset dalam proyek NLP dan bagaimana memanfaatkannya secara efisien.
2. Memanfaatkan library Datasets untuk mengelola dataset.
3. Belajar memuat, mengeksplorasi, dan memproses dataset NLP.
4. Menggunakan dataset untuk pelatihan model dan evaluasi.

1. Apa Itu Datasets Library?

Datasets adalah library open-source yang memudahkan akses dan pengelolaan dataset NLP. Fitur utama dari library ini adalah:

- Mendukung ratusan dataset publik.
- Dukungan format seperti CSV, JSON, dan Pandas.
- Kemampuan memproses dataset besar secara efisien menggunakan Apache Arrow.

2. Instalasi dan Import Library

Instalasi

pip install datasets

Import

from datasets import load_dataset

3. Memuat Dataset

Dataset Publik

Library ini menyediakan banyak dataset populer seperti GLUE, SQuAD, IMDB, dan lainnya.

dataset = load_dataset("imdb")

print(dataset)

Dataset Lokal

Jika Anda memiliki dataset lokal, Anda dapat memuatnya:

dataset = load_dataset("csv", data_files="path/to/your.csv")

4. Eksplorasi Dataset

Melihat Contoh Data

print(dataset["train"][0])

Statistik Dataset

print(dataset["train"].features)

print(dataset["train"].num_rows)

Membagi Dataset

Beberapa dataset memiliki pembagian bawaan:

- **train**
- **validation**
- **test**

train_dataset = dataset["train"]

test_dataset = dataset["test"]

5. Manipulasi Dataset

Filter Data

Untuk menyaring data berdasarkan kondisi:

filtered_dataset = dataset.filter(lambda x: x["label"] == 1)

Map Function

Memodifikasi dataset menggunakan fungsi map:

```
def tokenize_function(example):
    return tokenizer(example["text"], truncation=True)
tokenized_dataset = dataset.map(tokenize_function)
```

Concatenate Dataset

Menggabungkan dataset:

from datasets import concatenate_datasets

full_dataset = concatenate_datasets([train_dataset, test_dataset])

Split Dataset

Membagi dataset menjadi beberapa bagian:

train_test = dataset["train"].train_test_split(test_size=0.2)

6. Format Dataset

Mengonversi dataset menjadi format yang kompatibel dengan PyTorch atau TensorFlow.

 # PyTorch

tokenized_dataset.set_format(type="torch", columns=["input_ids", "attention_mask", "labels"])

 # TensorFlow

tokenized_dataset.set_format(type="tensorflow", columns=["input_ids", "attention_mask", "labels"])

7. Menyimpan dan Memuat Dataset

Untuk menyimpan dataset lokal:

dataset.save_to_disk("path/to/save")

Memuat kembali dataset:

from datasets import load_from_disk

dataset = load_from_disk("path/to/save")

8. Evaluasi Dataset

Library ini menyediakan metrik evaluasi yang kompatibel dengan dataset.

from datasets import load_metric

metric = load_metric("accuracy")

results = metric.compute(predictions=[0, 1], references=[0, 1])

print(results)

9. Studi Kasus: IMDB Dataset

Langkah-Langkah:

1. **Memuat Dataset**:

   dataset = load_dataset("imdb")

2. **Tokenisasi**:

   tokenized_dataset = dataset.map(lambda x: tokenizer(x["text"], truncation=True), batched=True)

3. **Pembagian Dataset**:

   train_test = tokenized_dataset["train"].train_test_split(test_size=0.2)

4. **Pelatihan Model**: Menggunakan dataset tokenized untuk melatih model.

Kesimpulan

Chapter 6 memberikan panduan lengkap tentang cara menggunakan library Datasets untuk memuat, memproses, dan mengevaluasi dataset NLP. Library ini dirancang untuk mempermudah akses ke dataset publik dan memberikan fleksibilitas dalam manipulasi data, sehingga sangat cocok untuk proyek-proyek NLP.