```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
pip install datasets
```

```
Collecting datasets
  Downloading datasets-3.0.1-py3-none-any.whl.metadata (20 kB)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from datasets) (3.16.1)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from datasets) (1.26.4)
Collecting pyarrow>=15.0.0 (from datasets)
  Downloading pyarrow-17.0.0-cp310-cp310-manylinux_2_28_x86_64.whl.metadata (3.3 kB)
Collecting dill<0.3.9,>=0.3.0 (from datasets)
  Downloading dill-0.3.8-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from datasets) (2.1.4)
Requirement already satisfied: requests>=2.32.2 in /usr/local/lib/python3.10/dist-packages (from datasets) (2.32.3)
Requirement already satisfied: tqdm>=4.66.3 in /usr/local/lib/python3.10/dist-packages (from datasets) (4.66.5)
Collecting xxhash (from datasets)
  Downloading xxhash-3.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (12 kB)
Collecting multiprocess (from datasets)
  Downloading multiprocess-0.70.17-py310-none-any.whl.metadata (7.2 kB)
Requirement already satisfied: fsspec<=2024.6.1,>=2023.1.0 in /usr/local/lib/python3.10/dist-packages (from fsspec[http]<=2024.6.1,>=202
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from datasets) (3.10.5)
Requirement already satisfied: huggingface-hub>=0.22.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (0.24.7)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from datasets) (24.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from datasets) (6.0.2)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (2.4.0)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (24.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (6.1.0)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.11.1)
Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (4.0.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.22.0->data
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets) (3.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets) (2024.8.3
INFO: pip is looking at multiple versions of multiprocess to determine which version is compatible with other requirements. This could t
  Downloading multiprocess-0.70.16-py310-none-any.whl.metadata (7.2 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2024.2)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2024.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas->datasets) (1.16
Downloading datasets-3.0.1-py3-none-any.whl (471 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 471.6/471.6 kB 14.0 MB/s eta 0:00:00
Downloading dill-0.3.8-py3-none-any.whl (116 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 116.3/116.3 kB 11.5 MB/s eta 0:00:00
Downloading pyarrow-17.0.0-cp310-cp310-manylinux_2_28_x86_64.whl (39.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 39.9/39.9 MB 32.0 MB/s eta 0:00:00
Downloading multiprocess-0.70.16-py310-none-any.whl (134 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.8/134.8 kB 13.1 MB/s eta 0:00:00
Downloading xxhash-3.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (194 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 194.1/194.1 kB 15.7 MB/s eta 0:00:00
Installing collected packages: xxhash, pyarrow, dill, multiprocess, datasets
  Attempting uninstall: pyarrow
    Found existing installation: pyarrow 14.0.2
    Uninstalling pyarrow-14.0.2:
      Successfully uninstalled pyarrow-14.0.2
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source
cudf-cu12 24.4.1 requires pyarrow<15.0.0a0,>=14.0.1, but you have pyarrow 17.0.0 which is incompatible.
Successfully installed datasets-3.0.1 dill-0.3.8 multiprocess-0.70.16 pyarrow-17.0.0 xxhash-3.5.0
```

Start coding or generate with AI.

```
from datasets import get_dataset_config_names

xtreme_subsets = get_dataset_config_names("xtreme")
print(f"XTREME has {len(xtreme_subsets)} configurations")
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:89: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secre
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
README.md: 100%                                       131k/131k [00:00<00:00, 2.45MB/s]
XTREME has 183 configurations
```

```
from datasets import DatasetDict , load_dataset
from collections import defaultdict
langs = ["de", "fr", "it", "en"]
fracs = [0.629, 0.229, 0.084, 0.059]
panx_ch = defaultdict(DatasetDict)

for lang, frac in zip(langs, fracs):
    ds = load_dataset("xtreme", name=f"PAN-X.{lang}")
    for split in ds:
        panx_ch[lang][split] = (
            ds[split]
            .shuffle(seed=0)
            .select(range(int(frac * ds[split].num_rows))))
```

```
train-00000-of-00001.parquet: 100%                    1.18M/1.18M [00:00<00:00, 5.03MB/s]
validation-00000-of-00001.parquet: 100%               590k/590k [00:00<00:00, 7.42MB/s]
test-00000-of-00001.parquet: 100%                     588k/588k [00:00<00:00, 7.01MB/s]
Generating train split: 100%                          20000/20000 [00:00<00:00, 78647.13 examples/s]
Generating validation split: 100%                     10000/10000 [00:00<00:00, 70136.78 examples/s]
Generating test split: 100%                           10000/10000 [00:00<00:00, 107342.58 examples/s]
train-00000-of-00001.parquet: 100%                    837k/837k [00:00<00:00, 3.60MB/s]
validation-00000-of-00001.parquet: 100%               419k/419k [00:00<00:00, 7.66MB/s]
test-00000-of-00001.parquet: 100%                     423k/423k [00:00<00:00, 7.78MB/s]
Generating train split: 100%                          20000/20000 [00:00<00:00, 273535.92 examples/s]
Generating validation split: 100%                     10000/10000 [00:00<00:00, 177034.61 examples/s]
Generating test split: 100%                           10000/10000 [00:00<00:00, 191114.94 examples/s]
train-00000-of-00001.parquet: 100%                    932k/932k [00:00<00:00, 8.45MB/s]
validation-00000-of-00001.parquet: 100%               459k/459k [00:00<00:00, 8.86MB/s]
test-00000-of-00001.parquet: 100%                     464k/464k [00:00<00:00, 5.80MB/s]
Generating train split: 100%                          20000/20000 [00:00<00:00, 44209.94 examples/s]
Generating validation split: 100%                     10000/10000 [00:00<00:00, 80251.64 examples/s]
Generating test split: 100%                           10000/10000 [00:00<00:00, 66070.41 examples/s]
train-00000-of-00001.parquet: 100%                    942k/942k [00:00<00:00, 4.10MB/s]
validation-00000-of-00001.parquet: 100%               472k/472k [00:00<00:00, 9.37MB/s]
test-00000-of-00001.parquet: 100%                     472k/472k [00:00<00:00, 9.66MB/s]
Generating train split: 100%                          20000/20000 [00:00<00:00, 167775.52 examples/s]
Generating validation split: 100%                     10000/10000 [00:00<00:00, 153169.03 examples/s]
Generating test split: 100%                           10000/10000 [00:00<00:00, 120979.30 examples/s]
```

```
tags = panx_ch["de"]["train"].features["ner_tags"].feature
```

```
tags
```

```
ClassLabel(names=['O', 'B-PER', 'I-PER', 'B-ORG', 'I-ORG', 'B-LOC', 'I-LOC'], id=None)
```

Start coding or generate with AI.

```
def create_tag_names(batch):
    return {"ner_tags_str": [tags.int2str(idx) for idx in batch["ner_tags"]]}

panx_de = panx_ch["de"].map(create_tag_names)
```

```
Map: 100%                                12580/12580 [00:04<00:00, 3887.95 examples/s]

Map: 100%                                6290/6290 [00:00<00:00, 8617.87 examples/s]

Map: 100%                                6290/6290 [00:00<00:00, 8356.55 examples/s]
```

```
from transformers import AutoTokenizer
```

```
The cache for model files in Transformers v4.22.0 has been updated. Migrating your old cache. This is a one-time only operation. You car
    0/0 [00:00<?, ?it/s]
```

```
xlmr_model_name = "xlm-roberta-base"


xlmr_tokenizer = AutoTokenizer.from_pretrained(xlmr_model_name)
```

```
tokenizer_config.json: 100%                                   25.0/25.0 [00:00<00:00, 1.39kB/s]

config.json: 100%                                    615/615 [00:00<00:00, 42.9kB/s]

sentencepiece.bpe.model: 100%                                   5.07M/5.07M [00:00<00:00, 27.1MB/s]

tokenizer.json: 100%                                   9.10M/9.10M [00:00<00:00, 19.6MB/s]

/usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils_base.py:1601: FutureWarning: `clean_up_tokenization_spaces` was
  warnings.warn(
```

```
!pip install torchcrf
```

```
Collecting torchcrf
  Downloading TorchCRF-1.1.0-py3-none-any.whl.metadata (2.3 kB)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from torchcrf) (1.26.4)
Requirement already satisfied: torch>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from torchcrf) (2.4.1+cu121)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch>=1.0.0->torchcrf) (3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.0.0->torchcrf) (4.12.2
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.0.0->torchcrf) (1.13.3)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.0.0->torchcrf) (3.3)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.0.0->torchcrf) (3.1.4)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch>=1.0.0->torchcrf) (2024.6.1)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.0.0->torchcrf) (2.1.5)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.0.0->torchcrf) (1.3.0
Downloading TorchCRF-1.1.0-py3-none-any.whl (5.2 kB)
Installing collected packages: torchcrf
Successfully installed torchcrf-1.1.0
```

```
import torch.nn as nn
from transformers import XLMRobertaConfig
from transformers.modeling_outputs import TokenClassifierOutput
from transformers.models.roberta.modeling_roberta import RobertaModel
from transformers.models.roberta.modeling_roberta import RobertaPreTrainedModel

class XLMRobertaForTokenClassification(RobertaPreTrainedModel):
    config_class = XLMRobertaConfig

    def __init__(self, config):
        super().__init__(config)
        self.num_labels = config.num_labels
        self.roberta = RobertaModel(config, add_pooling_layer=False)
        self.dropout = nn.Dropout(config.hidden_dropout_prob)
        self.classifier = nn.Linear(config.hidden_size, config.num_labels)
        self.init_weights()

    def forward(self, input_ids=None, attention_mask=None, token_type_ids=None,
                labels=None, **kwargs):
        outputs = self.roberta(input_ids, attention_mask=attention_mask,
                               token_type_ids=token_type_ids, **kwargs)
        sequence_output = self.dropout(outputs[0])
        logits = self.classifier(sequence_output)
        loss = None
        if labels is not None:
            loss_fct = nn.CrossEntropyLoss()
            loss = loss_fct(logits.view(-1, self.num_labels), labels.view(-1))
        return TokenClassifierOutput(loss=loss, logits=logits,
                                     hidden_states=outputs.hidden_states,
                                     attentions=outputs.attentions)
```

```
index2tag = {idx: tag for idx, tag in enumerate(tags.names)}
tag2index = {tag: idx for idx, tag in enumerate(tags.names)}
```

```
from transformers import AutoConfig

xlmr_config = AutoConfig.from_pretrained(xlmr_model_name,
                                         num_labels=tags.num_classes,
                                         id2label=index2tag, label2id=tag2index)
```

```
import torch

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
xlmr_model = (XLMRobertaForTokenClassification
              .from_pretrained(xlmr_model_name, config=xlmr_config)
              .to(device))
print(device)
```

```
model.safetensors: 100%                                   1.12G/1.12G [00:15<00:00, 63.0MB/s]
Some weights of XLMRobertaForTokenClassification were not initialized from the model checkpoint at xlm-roberta-base and are newly initia
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
cuda
```

```
def tag_text(text, tags, model, tokenizer):
    tokens = tokenizer(text).tokens()
    input_ids = xlmr_tokenizer(text, return_tensors="pt").input_ids.to(device)
    outputs = model(input_ids)[0]
    predictions = torch.argmax(outputs, dim=2)
    preds = [tags.names[p] for p in predictions[0].cpu().numpy()]
    return pd.DataFrame([tokens, preds], index=["Tokens", "Tags"])
```

```
def tokenize_and_align_labels(examples):
    tokenized_inputs = xlmr_tokenizer(examples["tokens"], truncation=True,
                                      is_split_into_words=True)
    labels = []
    for idx, label in enumerate(examples["ner_tags"]):
        word_ids = tokenized_inputs.word_ids(batch_index=idx)
        previous_word_idx = None
        label_ids = []
        for word_idx in word_ids:
            if word_idx is None or word_idx == previous_word_idx:
                label_ids.append(-100)
            else:
                label_ids.append(label[word_idx])
            previous_word_idx = word_idx
        labels.append(label_ids)
    tokenized_inputs["labels"] = labels
    return tokenized_inputs
```

```
def encode_panx_dataset(corpus):
    return corpus.map(tokenize_and_align_labels, batched=True,
                      remove_columns=['langs', 'ner_tags', 'tokens'])
```

```
panx_de_encoded = encode_panx_dataset(panx_ch["de"])
```

```
Map: 100%                                   12580/12580 [00:03<00:00, 2671.04 examples/s]
Map: 100%                                   6290/6290 [00:02<00:00, 2509.42 examples/s]
Map: 100%                                   6290/6290 [00:02<00:00, 2613.42 examples/s]
```

```
import numpy as np

def align_predictions(predictions, label_ids):
    preds = np.argmax(predictions, axis=2)
    batch_size, seq_len = preds.shape
    labels_list, preds_list = [], []

    for batch_idx in range(batch_size):
        example_labels, example_preds = [], []
        for seq_idx in range(seq_len):
            if label_ids[batch_idx, seq_idx] != -100:
                example_labels.append(index2tag[label_ids[batch_idx][seq_idx]])
                example_preds.append(index2tag[preds[batch_idx][seq_idx]])

        labels_list.append(example_labels)
        preds_list.append(example_preds)
    return preds_list, labels_list
```

```
pip install seqeval
```

```
Collecting seqeval
  Downloading seqeval-1.2.2.tar.gz (43 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 43.6/43.6 kB 3.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: numpy>=1.14.0 in /usr/local/lib/python3.10/dist-packages (from seqeval) (1.26.4)
Requirement already satisfied: scikit-learn>=0.21.3 in /usr/local/lib/python3.10/dist-packages (from seqeval) (1.5.2)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.21.3->seqeval) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.21.3->seqeval) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.21.3->seqeval) (3.5
Building wheels for collected packages: seqeval
  Building wheel for seqeval (setup.py) ... done
  Created wheel for seqeval: filename=seqeval-1.2.2-py3-none-any.whl size=16161 sha256=0e00f76df997f44517b56a442a4fb108b57d8724b960e9080
  Stored in directory: /root/.cache/pip/wheels/1a/67/4a/ad4082dd7dfc30f2abfe4d80a2ed5926a506eb8a972b4767fa
Successfully built seqeval
Installing collected packages: seqeval
Successfully installed seqeval-1.2.2
```

```
from seqeval.metrics import f1_score

def compute_metrics(eval_pred):
    y_pred, y_true = align_predictions(eval_pred.predictions,
                                       eval_pred.label_ids)
    return {"f1": f1_score(y_true, y_pred)}
```

```
from transformers import DataCollatorForTokenClassification

data_collator = DataCollatorForTokenClassification(xlmr_tokenizer)
```

```
def model_init():
    return (XLMRobertaForTokenClassification
            .from_pretrained(xlmr_model_name, config=xlmr_config)
            .to(device))
```

```
from transformers import TrainingArguments

num_epochs = 7
batch_size = 24
logging_steps = len(panx_de_encoded["train"]) // batch_size
model_name = f"{xlmr_model_name}-finetuned-panx-ner-1"
training_args = TrainingArguments(
    output_dir=model_name, log_level="error", num_train_epochs=num_epochs,
    per_device_train_batch_size=batch_size,
    per_device_eval_batch_size=batch_size, evaluation_strategy="epoch",
    save_steps=1e6, weight_decay=0.01, disable_tqdm=False,
    logging_steps=logging_steps, push_to_hub=True)
```

```
/usr/local/lib/python3.10/dist-packages/transformers/training_args.py:1525: FutureWarning: `evaluation_strategy` is deprecated and will
  warnings.warn(
from huggingface_hub import notebook_login

notebook_login()
```

Token is valid (permission: write).

Your token has been saved in your configured git credential helpers (store).

Your token has been saved to /root/.cache/huggingface/token

```
from transformers import Trainer

trainer = Trainer(model_init=model_init, args=training_args,
                  data_collator=data_collator, compute_metrics=compute_metrics,
                  train_dataset=panx_de_encoded["train"],
                  eval_dataset=panx_de_encoded["validation"],
                  tokenizer=xlmr_tokenizer)


trainer.train()
trainer.save_model("./content/drive")
trainer.push_to_hub(commit_message="Training completed!")
```

[3675/3675 19:43, Epoch 7/7]

Epoch	Training Loss	Validation Loss	F1
1	0.262200	0.154092	0.824247
2	0.138500	0.149915	0.840834
3	0.095400	0.157618	0.848817
4	0.065700	0.150909	0.860725
5	0.044800	0.163912	0.867715
6	0.030300	0.176465	0.869961
7	0.019200	0.182720	0.876781

```
events.out.tfevents.1727794443.fc5000bde54b.343.0: 100%                    9.32k/9.32k [00:00<00:00, 55.8kB/s]
```

```
No files have been modified since last commit. Skipping to prevent empty commit.
WARNING:huggingface_hub.hf_api:No files have been modified since last commit. Skipping to prevent empty commit.
CommitInfo(commit_url='https://huggingface.co/Khushiee/xlm-roberta-base-finetuned-panx-ner-
1/commit/6e4cb0b74e3fb85599d1aeeb9b1b6574b49331f5', commit_message='Training completed!', commit_description='',
oid='6e4cb0b74e3fb85599d1aeeb9b1b6574b49331f5', pr_url=None, pr_revision=None, pr_num=None)
```

```
text_de = "Google is located in London ."
tag_text(text_de, tags, trainer.model, xlmr_tokenizer)
```

	0	1	2	3	4	5	6	7	8
Tokens	<s>	_Google	_is	_located	_in	_London	_	.	</s>
Tags	O	B-ORG	O	O	O	B-LOC	O	O	O

Start coding or generate with AI.