lighteval is your AI evaluation library¶

This notebook explores how you can use lighteval to evaluate and compare LLMs.
lighteval has been around a while and it's a great tool for getting eval score on major benchmarks. It's just been refactored to support being used like a library in Python, which makes it great for comparing models across benchmarks.
So let's dig in to some eval scores.

Install dependencies

In [ ]:

```
!pip install -qqq -U "torch<2.5" "torchvision<2.5" "torchaudio<2.5" --index-url https://download.pytorch.org/whl/cu121
!pip list | grep torch
```

In [ ]:

```
!pip install -qqq git+https://github.com/huggingface/lighteval.git tiktoken
```

Setup lighteval evaluation

In [ ]:

```
import lighteval
from datetime import timedelta
from transformers import AutoModelForCausalLM

from lighteval.logging.evaluation_tracker import EvaluationTracker
from lighteval.pipeline import EnvConfig, ParallelismManager, Pipeline, PipelineParameters
from kaggle_secrets import UserSecretsClient
user_secrets = UserSecretsClient()
TOKEN = user_secrets.get_secret("HF_TOKEN")
```

In [ ]:

```
env_config = EnvConfig(token=TOKEN, cache_dir="~/tmp")

evaluation_tracker = EvaluationTracker(
    output_dir="~/tmp",
    save_details=False,
    push_to_hub=False,
    push_to_tensorboard=False,
    public=False,
    hub_results_org=False,
)
```

```
pipeline_params = PipelineParameters(
    launcher_type=ParallelismManager.ACCELERATE,
    env_config=env_config,
    job_id=1,
    override_batch_size=1,
    num_fewshot_seeds=0,
    max_samples=10,
    use_chat_template=False,
)
```

In [ ]:

```
domain_tasks = "leaderboard|mmlu:anatomy|5|0"
```

In [ ]:

```
qwen_model = AutoModelForCausalLM.from_pretrained("Qwen/
Qwen2.5-0.5B")

pipeline = Pipeline(
    tasks=domain_tasks,
    pipeline_parameters=pipeline_params,
    evaluation_tracker=evaluation_tracker,
    model=qwen_model
)

pipeline.evaluate()

qwen_results = pipeline.get_results()
```

In [ ]:

```
pipeline.show_results()
```

In [ ]:

```
smol_model =
AutoModelForCausalLM.from_pretrained("HuggingFaceTB/
SmolLM2-360M-Instruct")

pipeline = Pipeline(
    tasks=domain_tasks,
    pipeline_parameters=pipeline_params,
    evaluation_tracker=evaluation_tracker,
    model=smol_model
)

pipeline.evaluate()
```

In [ ]:

```
smol_results = pipeline.get_results()
```

In [ ]:

```
pipeline.show_results()
```

In [ ]:

```
import pandas as pd

df =
pd.DataFrame.from_records(smol_results["results"]).T["acc"].rename
("SmolLM2-360M-Instruct")
_df =
pd.DataFrame.from_records(qwen_results["results"]).T["acc"].renam
e("Qwen2-0.5B-DPO")
df = pd.concat([df, _df], axis=1)
df.plot(kind="barh")
```

In [ ]: