Hugging Face Model Testing and Migration Script

Overview

This script helps test and migrate models from Hugging Face to a local repository, specifically for the LLM chakra architecture components.

```
import os from huggingface_hub import HfApi, Repository from
transformers import AutoModel, AutoTokenizer import torch #
Configure model mappings CHAKRA_MODELS = { 'smriti': 'intfloat/e5-
base-v2', 'brahman': 'meta-llama/llama-2-7b', 'buddhi': 'deepseek-
ai/deepseek-coder-6.7b-base', 'karma': 'openchat/openchat-3.5',
'jnana': 'mistralai/Mistral-7B-Instruct-v0.2', 'vahana':
'teknium/Hermes-2-Pro-Mistral-7B', 'raksha':
'togethercomputer/RedPajama-INCITE-Base-7B-v0.1', 'atman':
'microsoft/phi-2' } def test_model(model_name, model_id): """Test if
model can be loaded and run basic inference""" try: print(f"Testing
{model_name}...") tokenizer =
AutoTokenizer.from_pretrained(model_id) model =
AutoModel.from_pretrained(model_id) # Basic inference test test_text
= "Testing model functionality." inputs = tokenizer(test_text,
return_tensors="pt") outputs = model(**inputs) print(f"✓
{model_name} loaded and tested successfully") return True except
Exception as e: print(f"✗ Error testing {model_name}: {str(e)}")
return False def clone_to_repo(model_id, local_dir): """Clone model
to local repository""" try: repo = Repository(local_dir=local_dir,
clone_from=model_id) print(f"Successfully cloned {model_id} to
{local_dir}") return True except Exception as e: print(f"Error
cloning {model_id}: {str(e)}") return False def main(): # Create
local repository directory base_dir = "./chakra_models"
os.makedirs(base_dir, exist_ok=True) results = [] for chakra_name,
model_id in CHAKRA_MODELS.items(): print(f"\nProcessing
{chakra_name.upper()} chakra model") # Test model test_success =
test_model(chakra_name, model_id) # Clone to local repository if
test successful local_dir = os.path.join(base_dir, chakra_name) if
test_success: clone_success = clone_to_repo(model_id, local_dir)
else: clone_success = False results.append({ 'chakra': chakra_name,
'model_id': model_id, 'test_success': test_success, 'clone_success':
clone_success }) # Print summary print("\n=== Migration Summary
===") for result in results: status = "✓" if result['test_success']
and result['clone_success'] else "✗" print(f"{status}
{result['chakra']}: {result['model_id']}") if __name__ ==
"__main__": main()
```

Usage Instructions