Text Classification with HuggingFace Transformers in FastAPI

This example demonstrates how to deploy a **transformer-based text classification model** (like BERT) using FastAPI. You'll be able to send text input and get back class predictions (e.g., sentiment analysis, spam detection, etc.).

Step 1: Install Dependencies

```bash
pip install fastapi uvicorn transformers torch
```

Step 2: Load a Pretrained HuggingFace Model

We'll use `distilbert-base-uncased-finetuned-sst-2-english` for sentiment classification (positive/negative).

```python
from fastapi import FastAPI
from pydantic import BaseModel
from transformers import AutoTokenizer, AutoModelForSequenceClassification
from transformers import pipeline

# Define FastAPI app
app = FastAPI()

# Load tokenizer and model
model_name = "distilbert-base-uncased-finetuned-sst-2-english"
tokenizer = AutoTokenizer.from_pretrained(model_name)
```

```
model = AutoModelForSequenceClassification.from_pretrained(model_name)

# Create a sentiment pipeline
classifier = pipeline("sentiment-analysis", model=model, tokenizer=tokenizer)

# Pydantic schema for request
class TextInput(BaseModel):
    text: str

@app.post("/predict/")
def predict_sentiment(input: TextInput):
    result = classifier(input.text)
    return {"label": result[0]['label'], "score": float(result[0]['score'])}
```

Step 3: Run the FastAPI Server

```bash
uvicorn main:app --reload
```

- Go to: `http://127.0.0.1:8000/docs`
- Use Swagger UI to test your `/predict/` endpoint.

Example Input and Output

Request:

```json
```

POST /predict/

{

  "text": "This movie was absolutely fantastic!"

}

```

**Response**:

```json

{

  "label": "POSITIVE",

  "score": 0.9998476505

}

```

Notes

- The `pipeline` API makes it easy to wrap inference.

- You can replace the model with any Hugging Face classification model (spam detection, hate speech, news categories, etc.)

- For performance: use `TorchScript` or `ONNX` for deployment.

Extensions

- Add batch prediction support

- Secure with API keys (e.g., for public web app)

- Dockerize and deploy on Render, HuggingFace Spaces, or AWS

```

With this setup, you have a production-ready text classifier using the power of Hugging Face transformers and FastAPI.