BUILDING MACHINE LEARNING DEMOS WITH PYTHON

Omar Sanseviero (osanseviero)
Interactive demo: Helsinki-NLP English to Spanish Translation

TEXT
Hello! My name is Omar

OUTPUT
Hola, mi nombre es Omar.

Examples
Hello! My name is Omar   I like this workshop
<table>
<thead>
<tr>
<th>TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hello! My name is Omar</td>
</tr>
</tbody>
</table>

**Examples**

Hello! My name is Omar
import gradio as gr

from transformers import pipeline

pipe = pipeline("translation", model="Helsinki-NLP/opus-mt-en-es")

def predict(text):
    return pipe(text)[0]["translation_text"]

title = "Interactive demo: Helsinki-NLP English to Spanish Translation"

iface = gr.Interface(
    fn=predict,
    inputs=[gr.inputs.Textbox(label="text", lines=3)],
    outputs="text",
    title=title,
    examples=[["Hello! My name is Omar"], ["I like this workshop"]]
)

iface.launch()
```python
import gradio as gr
from transformers import pipeline

pipe = pipeline("translation", model="Helsinki-NLP/opus-mt-en-es")

def predict(text):
    return pipe(text)[0]["translation_text"]

title = "Interactive demo: Helsinki-NLP English to Spanish Translation"

iface = gr.Interface(
    fn=predict,
    inputs=[gr.inputs.Textbox(label="text", lines=3)],
    outputs='text',
    title=title,
    examples=["Hello! My name is Omar", ["I like this workshop"]]
)

iface.launch()
```
import gradio as gr

from transformers import pipeline

pipe = pipeline("translation", model="Helsinki-NLP/opus-mt-en-es")

def predict(text):
    return pipe(text)[0]["translation_text"]

title = "Interactive demo: Helsinki-NLP English to Spanish Translation"

iface = gr.Interface(
    fn=predict,
    inputs=[gr.inputs.Textbox(label="text", lines=3)],
    outputs='text',
    title=title,
    examples=["Hello! My name is Omar"], ["I like this workshop"]
)

iface.launch()
import gradio as gr

from transformers import pipeline

pipe = pipeline("translation", model="Helsinki-NLP/opus-mt-en-es")

def predict(text):
    return pipe(text)[0]["translation_text"]

title = "Interactive demo: Helsinki-NLP English to Spanish Translation"

iface = gr.Interface(
    fn=predict,
    inputs=[gr.inputs.Textbox(label="text", lines=3)],
    outputs='text',
    title=title,
    examples=[['Hello! My name is Omar'], ['I like this workshop']]
)

iface.launch()
<table>
<thead>
<tr>
<th>TEXT</th>
<th>OUTPUT</th>
<th>0.46s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hello! My name is Omar</td>
<td>Hola, mi nombre es Omar.</td>
<td></td>
</tr>
</tbody>
</table>

**Examples**

<table>
<thead>
<tr>
<th>English</th>
<th>Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hello! My name is Omar</td>
<td>Hola, mi nombre es Omar.</td>
</tr>
<tr>
<td>I like this workshop</td>
<td></td>
</tr>
</tbody>
</table>
Always has been

Wait Gradio is all
React under the hood??
Gradio vs Streamlit

- **Gradio**
  - Build extremely easy to use interfaces
  - Only specify input and outputs
- **Streamlit**
  - For flexible to a fully-fledged web apps
  - Community components

Both are brilliant!
I built a ML demo

Can you share with the community?

Let’s not get ahead of ourselves.
hf.co/spaces/
hf.co/spaces/to-the-moon
Create a new Space
A repository contains all space files, including the revision history.

Owner

Space name

Select the Space SDK
You can choose between Streamlit and Gradio for your Space. Contact us if you need a custom solution.

Streamlit
Gradio
Add yours

Public
Anyone on the internet can see this space. Only you (personal space) or members of your organization (organization space) can commit.

Private
Only you (personal space) or members of your organization (organization space) can see and commit to this space.

Create space
Use case 1. Load and execute models
Use case 1. Load and execute models
Use case 2. Use Inference API
Use case 2. Use Inference API

```python
import gradio as gr

description = "BigGAN text-to-image demo."
title = "BigGAN ImageNet"

interface = gr.Interface.load(
    "huggingface/osanseviero/BigGAN-deep-128",
    description=description,
    title=title,
    examples=[["american robin"], ["chest"], ["soap bubble"]]
)

interface.launch()
```
Use case 3. Series of Models

- Spanish to English
- Generate text
- English to Spanish
Use case 3. Series of Models

```python
import gradio as gr
from gradio.mix import Series

description = "Generate your own D&D story!"
title = "Spanish Story Generator using Opus MT and GPT-2"

translator_es = gr.Interface.load("huggingface/Helsinki-NLP/opus-mt-es-en")
story_gen = gr.Interface.load("huggingface/pranavpsv/gpt2-genre-story-generator")
translator_en = gr.Interface.load("huggingface/Helsinki-NLP/opus-mt-en-es")

examples = [["La aventura comienza en"]]

interface = Series(translator_es, story_gen, translator_en, description = description, title = title,
                   examples=examples,
                   inputs = gr.inputs.Textbox(lines = 10))

interface.launch()
```
Selected Spaces

- DALLE Mini
- BERTIN
- DOCTR
- Spacy Visualization
- HUBERT
THANK YOU!

Omar Sanseviero (osanseviero)