

# Session 1 — Your First Space

## Space Builder: Silly Phrase Finder

### What You'll Build

A Space that takes any paragraph of text and uses AI to find the silliest phrase in it. It scores each sentence for silliness and shows you the winner.

### Step 1 — Create a New Space

1. Go to [huggingface.co/new-space](https://huggingface.co/new-space) (log in if needed)
2. In the "Space name" field, type: silly-phrase-finder
3. Under "Select the Space SDK", choose Gradio
4. Under "Select the Space hardware", choose Free — CPU basic
5. Leave everything else as default
6. Click Create Space

### Step 2 — Create the requirements.txt File

1. On your Space page, click the Files tab (near the top)
2. Click Add file → Create a new file
3. In the filename field at the top, type: requirements.txt
4. In the big text area below, copy and paste this exactly:

*Your Space page will open. It's empty right now — we're about to add the code.*

5. Click Commit new file to main (the blue button at the bottom)

### Step 3 — Create the app.py File

1. Click Add file → Create a new file (again)
2. In the filename field, type: app.py
3. In the big text area, copy and paste ALL of the code below:

```
transformers
torch
gradio
```

4. Click Commit new file to main

## **Step 4 — Wait for It to Build**

1. Click the App tab (at the top of your Space page)
2. You'll see "Building" — this takes 2–5 minutes
3. When it's done, your Space will appear!

## **Step 5 — Try It Out!**

1. Paste a paragraph from a news article mixed with something silly
2. Try your favorite book passage — does the AI agree on what's silly?
3. Write your own paragraph with one obviously silly sentence hidden in it

## **Troubleshooting**

```

import gradio as gr
from transformers import pipeline
import re

# Load a zero-shot classification model (works on free CPU)
classifier = pipeline("zero-shot-classification", model="valhalla/distilbart-mnli-12-3")

def find_silliest(text):
    if not text or not text.strip():
        return "Paste some text above first!"

    # Split on sentence-ending punctuation followed by whitespace
    sentences = [s.strip() for s in re.split(r'(?<=[!?!])\s+(?=[A-Z])', text) if len(s.strip()) > 30]

    if len(sentences) == 0:
        return "I need at least a couple of sentences to compare. Paste a longer passage!"

    # Score every sentence for "silly" vs "serious" vs "ordinary"
    labels = ["silly and ridiculous", "serious and important", "ordinary and boring"]
    results = classifier(sentences, candidate_labels=labels)

    # Handle single sentence (returns dict instead of list)
    if isinstance(results, dict):
        results = [results]

    # Find the sentence with the highest "silly" score
    best_phrase = ""
    best_score = 0
    for sentence, result in zip(sentences, results):
        silly_idx = result["labels"].index("silly and ridiculous")
        score = result["scores"][silly_idx]
        if score > best_score:
            best_score = score
            best_phrase = sentence

    return f'"{best_phrase}"\n\nSilliness score: {best_score:.0%}'

demo = gr.Interface(
    fn=find_silliest,
    inputs=gr.Textbox(lines=10, placeholder="Paste a paragraph or two here..."),
    outputs=gr.Textbox(label="The Silliest Phrase"),
    title="Silly Phrase Finder",
    description="Paste any text and this AI will pick out the silliest phrase. It uses a zero-shot classifier – no one ever trained it on 'silly,' it just figures it out!",
    examples=[
        ["The quarterly budget report is due on Friday. My cat learned to open the refrigerator and now judges my food choices. Please remember to submit your timesheets. The printer on the third floor is out of toner again."],
        ["The sun rose over the mountains. A penguin wearing a tiny hat skateboarded through the library. Students studied quietly. The teacher handed out assignments."],
    ],
)

demo.launch()

```

Make sure you copy the ENTIRE code block – from the very first line to the very last. Missing even one line can cause errors.

If you see a red error: Click the Logs tab to read the error message. The most common fix is to double-check that requirements.txt has the right contents and that app.py was copied completely.

Problem	Fix
"Runtime error"	Check the Logs tab. Usually means a typo in app.py. Re-copy the code carefully.
Space stuck on "Building"	Wait up to 5 minutes. Free CPU Spaces can be slow. If it's been more than 10 minutes, try deleting the Space and starting over.
"ModuleNotFoundError"	Your requirements.txt is missing a library. Make sure it matches exactly what's shown above.
Space loads but nothing happens	Make sure the last line of app.py is demo.launch() with no extra spaces before it.