```
!pip install torch torchvision torchaudio
!pip install transformers ipywidgets gradio --upgrade
!pip install sentencepiece
pip install sacremoses

import gradio as gr
from transformers import pipeline

translation_pipelines = {
    "German": pipeline(task="translation", model="Helsinki-NLP/opus-mt-en-de"),
    "French": pipeline(task="translation", model="Helsinki-NLP/opus-mt-en-fr"),
    "Romanian": pipeline(task="translation", model="Helsinki-NLP/opus-mt-en-ro"),
}

def translate_to_german(text):
    try:
        translation_result = translation_pipelines["German"](text)
        return translation_result[0]['translation_text']
    except Exception as e:
        return f"Error during translation: {str(e)}"

def translate_to_french(text):
    try:
        translation_result = translation_pipelines["French"](text)
        return translation_result[0]['translation_text']
    except Exception as e:
        return f"Error during translation: {str(e)}"

def translate_to_romanian(text):
    try:
        translation_result = translation_pipelines["Romanian"](text)
        return translation_result[0]['translation_text']
    except Exception as e:
        return f"Error during translation: {str(e)}"

def translate_transformers(text, target_language):
    if target_language == "German":
        return translate_to_german(text)
    elif target_language == "French":
        return translate_to_french(text)
    elif target_language == "Romanian":
        return translate_to_romanian(text)
    else:
        return "Invalid target language"

languages = [("German", "German"), ("French", "French"), ("Romanian", "Romanian")]

interface = gr.Interface(
    fn=translate_transformers,
    inputs=[gr.Textbox(lines=2, placeholder='Text to translate'), gr.Dropdown(languages, label="Target Language")],
    outputs=gr.Textbox(),  # Use Gradio Textbox for output
    live=True,
    title='Language Translation App',
    description='Translate text from English to the selected language.'
)

interface.launch()
```