```
!pip install customtkinter
!pip install diffusers
```

```
Collecting customtkinter
  Downloading customtkinter-5.2.2-py3-none-any.whl (296 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 296.1/296.1 kB 4.4 MB/s eta 0:00:00
Collecting darkdetect (from customtkinter)
  Downloading darkdetect-0.8.0-py3-none-any.whl (9.0 kB)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from customtkinter) (24.0)
Installing collected packages: darkdetect, customtkinter
Successfully installed customtkinter-5.2.2 darkdetect-0.8.0
Collecting diffusers
  Downloading diffusers-0.27.2-py3-none-any.whl (2.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 8.8 MB/s eta 0:00:00
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.10/dist-packages (from diffusers) (7.1.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from diffusers) (3.13.4)
Requirement already satisfied: huggingface-hub>=0.20.2 in /usr/local/lib/python3.10/dist-packages (from diffusers) (0.20.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from diffusers) (1.25.2)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from diffusers) (2023.12.25)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from diffusers) (2.31.0)
Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.10/dist-packages (from diffusers) (0.4.3)
Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages (from diffusers) (9.4.0)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.20.2->diffusers) (2023.6.0)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.20.2->diffusers) (4.66.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.20.2->diffusers) (6.0.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.20.2->diffusers) (4.11.0)
Requirement already satisfied: packaging>=20.9 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.20.2->diffusers) (24.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.10/dist-packages (from importlib-metadata->diffusers) (3.18.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->diffusers) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->diffusers) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->diffusers) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->diffusers) (2024.2.2)
Installing collected packages: diffusers
Successfully installed diffusers-0.27.2
```

```
import tkinter as tk
import customtkinter as ctk
from PIL import ImageTk
import torch
from torch import autocast
from diffusers import StableDiffusionPipeline
from IPython.display import Image, display

ctk.set_appearance_mode("Dark")  # Modes: "System" (standard), "Dark", "Light"
ctk.set_default_color_theme("dark-blue")  # Themes: "blue" (standard), "green", "dark-blue"

class ImageGeneratorApp(ctk.CTk):
    def __init__(self):
        super().__init__()

        # Configures window
        self.default_window_width = 1200
        self.default_window_height = 800

        # Default values
        self.authorization_token = ""
        self.modelid = "CompVis/stable-diffusion-v1-4"
        self.device = "cuda"

        self.title("Image Generator")
        self.geometry(f"{self.default_window_width}x{self.default_window_height}")

        # Generates user interface
        self.create_widgets()

    def create_widgets(self):
        self.window_label = ctk.CTkLabel(self, text="Image Generator", font=ctk.CTkFont(size=30, weight="bold"), padx=50, pady=50, text_color="white")
        self.window_label.pack()

        self.prompt_label = ctk.CTkLabel(self, text="Prompt", font=ctk.CTkFont(family="Times New Roman", size=20, weight="bold"), text_color="white")
        self.prompt_label.pack()

        self.prompt_entry = ctk.CTkEntry(self, placeholder_text="Enter your prompt here", width=self.default_window_width-20, height=40)
        self.prompt_entry.pack(padx=20, pady=20)

        self.generate_button = ctk.CTkButton(master=self, text="Generate Image", width=self.default_window_width-50, height=40, fg_color="transparent", border
        self.generate_button.pack()

    def generate(self):
        text_prompt = self.prompt_entry.get()

        self.generate_button.configure(state="disabled")

        progress = ctk.CTkProgressBar(master=self, orientation='horizontal', mode='indeterminate')
        progress.pack()
        progress.start()

        pipeline = StableDiffusionPipeline.from_pretrained(self.modelid, revision="fp16", torch_dtype=torch.float16, use_auth_token=self.authorization_token)
        pipeline.to(self.device)

        with autocast():
            generated_image = pipeline(text_prompt, guidance_scale=8.5).images[0]
            generated_image.save('generated_image.png')

        progress.stop()
        progress.pack_forget()
        self.generate_button.configure(state="normal")

        # Display the generated image in Colab
        display(Image(filename='generated_image.png'))


import torch
from torch import autocast
from diffusers import StableDiffusionPipeline
import matplotlib.pyplot as plt


authorization_token = ""
modelid = "CompVis/stable-diffusion-v1-4"
device = "cuda"


pipe = StableDiffusionPipeline.from_pretrained(modelid, revision="fp16", torch_dtype=torch.float16, use_auth_token=authorization_token)

pipe.to(device)
```

```
Cannot initialize model with low cpu memory usage because `accelerate` was not found in the environm
```
```
pip install accelerate
```
```
.
safety_checker/model.safetensors not found
Keyword arguments {'use_auth_token': ''} are not expected by StableDiffusionPipeline and will be ign
```

Loading pipeline components...: 100% 7/7 [00:25<00:00, 3.15s/it]


```
StableDiffusionPipeline {
  "_class_name": "StableDiffusionPipeline",
  "_diffusers_version": "0.27.2",
  "_name_or_path": "CompVis/stable-diffusion-v1-4",
  "feature_extractor": [
    "transformers",
    "CLIPFeatureExtractor"
  ],
  "image_encoder": [
    null,
    null
  ],
  "requires_safety_checker": true,
  "safety_checker": [
    "stable_diffusion",
    "StableDiffusionSafetyChecker"
  ],
  "scheduler": [
    "diffusers",
    "PNDMScheduler"
  ],
  "text_encoder": [
    "transformers",
    "CLIPTextModel"
  ],
  "tokenizer": [
    "transformers",
    "CLIPTokenizer"
  ],
  "unet": [
    "diffusers",
    "UNet2DConditionModel"
  ],
  "vae": [
    "diffusers",
    "AutoencoderKL"
  ]
}
```

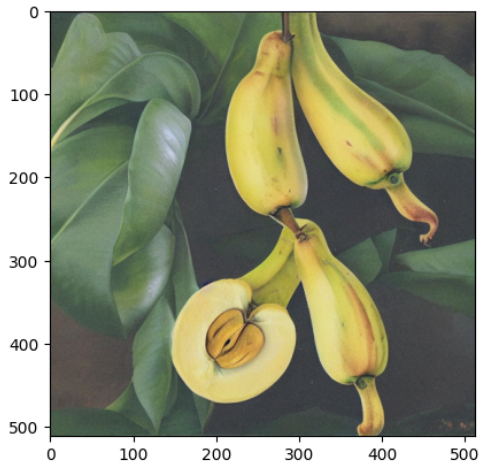

```
with autocast(device):
  textprompt = str(input("Enter your prompt: "))

  image = pipe(textprompt, guidance_scale=8.5).images[0]

  imgplot = plt.imshow(image)
```

```
Enter your prompt: Apple and banana
```

100% 50/50 [00:08<00:00, 6.37it/s]

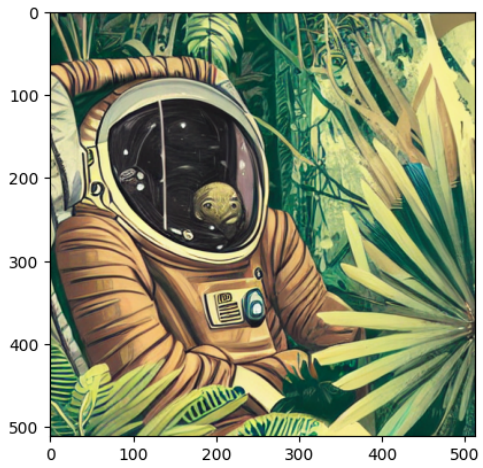



```
with autocast(device):
  textprompt = str(input("Enter your prompt: "))

  image = pipe(textprompt, guidance_scale=8.5).images[0]

  imgplot = plt.imshow(image)
```

```
Enter your promt: Astronaut in a jungle, cold color palette, muted colors, detailed, 8k
100%                                                 50/50 [00:08<00:00,  6.31it/s]
```

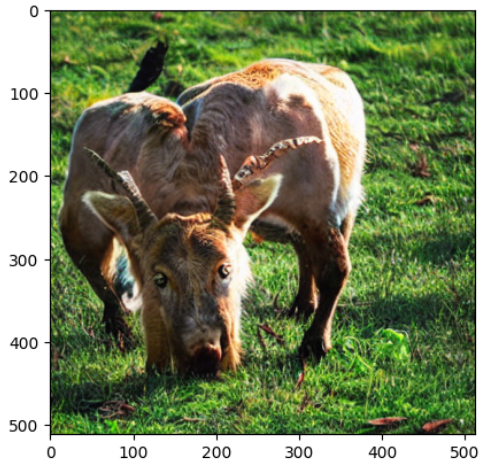


```
with autocast(device):
  textprompt = str(input("Enter your prompt: "))

  image = pipe(textprompt, guidance_scale=8.5).images[0]

  imgplot = plt.imshow(image)
```

```
Enter your promt: animal eating grass
100%                                                 50/50 [00:09<00:00,  5.38it/s]
```

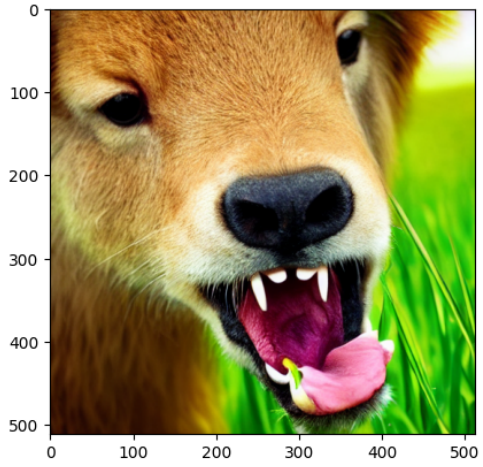


```
with autocast(device):
  textprompt = str(input("Enter your prompt: "))

  image = pipe(textprompt, guidance_scale=8.5).images[0]


  imgplot = plt.imshow(image)
```

```
Enter your promt: animal eating grass, detailed, 8k
100%                                                 50/50 [00:08<00:00,  6.19it/s]
```

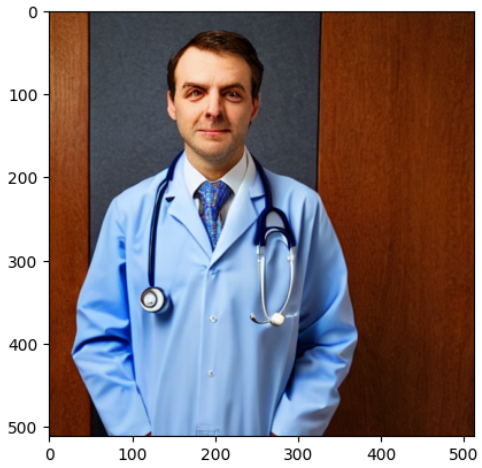


```
with autocast(device):
  textprompt = str(input("Enter your prompt: "))

  image = pipe(textprompt, guidance_scale=8.5).images[0]

  imgplot = plt.imshow(image)
```

```
Enter your promt: Doctor
100%                                                 50/50 [00:08<00:00,  6.23it/s]
```




```
!pip install salesforce-lavis
```

```
Collecting salesforce-lavis
  Downloading salesforce_lavis-1.0.2-py3-none-any.whl (1.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 18.1 MB/s eta 0:00:00
Collecting contexttimer (from salesforce-lavis)
  Downloading contexttimer-0.3.3.tar.gz (4.9 kB)
  Preparing metadata (setup.py) ... done
Collecting decord (from salesforce-lavis)
  Downloading decord-0.6.0-py3-none-manylinux2010_x86_64.whl (13.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.6/13.6 MB 62.1 MB/s eta 0:00:00
Collecting einops>=0.4.1 (from salesforce-lavis)
  Downloading einops-0.8.0-py3-none-any.whl (43 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 43.2/43.2 kB 4.7 MB/s eta 0:00:00
Collecting fairscale==0.4.4 (from salesforce-lavis)
  Downloading fairscale-0.4.4.tar.gz (235 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 235.4/235.4 kB 24.1 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
  Preparing metadata (pyproject.toml) ... done
Collecting ftfy (from salesforce-lavis)
  Downloading ftfy-6.2.0-py3-none-any.whl (54 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 54.4/54.4 kB 7.8 MB/s eta 0:00:00
Collecting iopath (from salesforce-lavis)
  Downloading iopath-0.1.10.tar.gz (42 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 42.2/42.2 kB 6.0 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: ipython in /usr/local/lib/python3.10/dist-packages (from salesforce-l
Collecting omegaconf (from salesforce-lavis)
  Downloading omegaconf-2.3.0-py3-none-any.whl (79 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 79.5/79.5 kB 11.7 MB/s eta 0:00:00
Collecting opencv-python-headless==4.5.5.64 (from salesforce-lavis)
  Downloading opencv_python_headless-4.5.5.64-cp36-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.w
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 47.8/47.8 MB 23.8 MB/s eta 0:00:00
Collecting opendatasets (from salesforce-lavis)
  Downloading opendatasets-0.1.22-py3-none-any.whl (15 kB)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from salesforce
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from salesforce-la
Requirement already satisfied: plotly in /usr/local/lib/python3.10/dist-packages (from salesforce-la
Collecting pre-commit (from salesforce-lavis)
  Downloading pre_commit-3.7.1-py2.py3-none-any.whl (204 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 204.3/204.3 kB 22.6 MB/s eta 0:00:00
Collecting pycocoevalcap (from salesforce-lavis)
  Downloading pycocoevalcap-1.2-py3-none-any.whl (104.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 104.3/104.3 MB 6.1 MB/s eta 0:00:00
Requirement already satisfied: pycocotools in /usr/local/lib/python3.10/dist-packages (from salesfor
Collecting python-magic (from salesforce-lavis)
  Downloading python_magic-0.4.27-py2.py3-none-any.whl (13 kB)
Requirement already satisfied: scikit-image in /usr/local/lib/python3.10/dist-packages (from salesfo
Requirement already satisfied: sentencepiece in /usr/local/lib/python3.10/dist-packages (from salesf
Requirement already satisfied: spacy in /usr/local/lib/python3.10/dist-packages (from salesforce-lav
Collecting streamlit (from salesforce-lavis)
  Downloading streamlit-1.34.0-py2.py3-none-any.whl (8.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.5/8.5 MB 73.6 MB/s eta 0:00:00
Collecting timm==0.4.12 (from salesforce-lavis)
  Downloading timm-0.4.12-py3-none-any.whl (376 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 377.0/377.0 kB 44.4 MB/s eta 0:00:00
Requirement already satisfied: torch>=1.10.0 in /usr/local/lib/python3.10/dist-packages (from salesf
Requirement already satisfied: torchvision in /usr/local/lib/python3.10/dist-packages (from salesfor
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from salesforce-lavi
Collecting transformers<4.27,>=4.25.0 (from salesforce-lavis)
  Downloading transformers-4.26.1-py3-none-any.whl (6.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.3/6.3 MB 98.5 MB/s eta 0:00:00
Collecting webdataset (from salesforce-lavis)
  Downloading webdataset-0.2.86-py3-none-any.whl (70 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 70.4/70.4 kB 11.6 MB/s eta 0:00:00
Requirement already satisfied: wheel in /usr/local/lib/python3.10/dist-packages (from salesforce-lav
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.10/dist-packages (from opencv
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch>=1.10
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0-
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.10
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0
Collecting nvidia-cuda-nvrtc-cu12==12.1.105 (from torch>=1.10.0->salesforce-lavis)
  Using cached nvidia_cuda_nvrtc_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (23.7 MB)
Collecting nvidia-cuda-runtime-cu12==12.1.105 (from torch>=1.10.0->salesforce-lavis)
  Using cached nvidia_cuda_runtime_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (823 kB)
Collecting nvidia-cuda-cupti-cu12==12.1.105 (from torch>=1.10.0->salesforce-lavis)
  Using cached nvidia_cuda_cupti_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (14.1 MB)
Collecting nvidia-cudnn-cu12==8.9.2.26 (from torch>=1.10.0->salesforce-lavis)
  Using cached nvidia_cudnn_cu12-8.9.2.26-py3-none-manylinux1_x86_64.whl (731.7 MB)
Collecting nvidia-cublas-cu12==12.1.3.1 (from torch>=1.10.0->salesforce-lavis)
  Using cached nvidia_cublas_cu12-12.1.3.1-py3-none-manylinux1_x86_64.whl (410.6 MB)
Collecting nvidia-cufft-cu12==11.0.2.54 (from torch>=1.10.0->salesforce-lavis)
  Using cached nvidia_cufft_cu12-11.0.2.54-py3-none-manylinux1_x86_64.whl (121.6 MB)
Collecting nvidia-curand-cu12==10.3.2.106 (from torch>=1.10.0->salesforce-lavis)
  Using cached nvidia_curand_cu12-10.3.2.106-py3-none-manylinux1_x86_64.whl (56.5 MB)
Collecting nvidia-cusolver-cu12==11.4.5.107 (from torch>=1.10.0->salesforce-lavis)
  Using cached nvidia_cusolver_cu12-11.4.5.107-py3-none-manylinux1_x86_64.whl (124.2 MB)
Collecting nvidia-cusparse-cu12==12.1.0.106 (from torch>=1.10.0->salesforce-lavis)
  Using cached nvidia_cusparse_cu12-12.1.0.106-py3-none-manylinux1_x86_64.whl (196.0 MB)
Collecting nvidia-nccl-cu12==2.19.3 (from torch>=1.10.0->salesforce-lavis)
  Using cached nvidia_nccl_cu12-2.19.3-py3-none-manylinux1_x86_64.whl (166.0 MB)
Collecting nvidia-nvtx-cu12==12.1.105 (from torch>=1.10.0->salesforce-lavis)
  Using cached nvidia_nvtx_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (99 kB)
Requirement already satisfied: triton==2.2.0 in /usr/local/lib/python3.10/dist-packages (from torch>
Collecting nvidia-nvjitlink-cu12 (from nvidia-cusolver-cu12==11.4.5.107->torch>=1.10.0->salesforce-l
  Using cached nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (21.1 MB)
Requirement already satisfied: huggingface-hub<1.0,>=0.11.0 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transfor
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from tr
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformer
Collecting tokenizers!=0.11.3,<0.14,>=0.11.1 (from transformers<4.27,>=4.25.0->salesforce-lavis)
  Downloading tokenizers-0.13.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (7.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.8/7.8 MB 77.2 MB/s eta 0:00:00
```

```
Requirement already satisfied: wcwidth<0.3.0,>=0.2.12 in /usr/local/lib/python3.10/dist-packages (fr
Collecting portalocker (from iopath->salesforce-lavis)
  Downloading portalocker-2.8.2-py3-none-any.whl (17 kB)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.10/dist-packages (from ipy
Collecting jedi>=0.16 (from ipython->salesforce-lavis)
  Downloading jedi-0.19.1-py2.py3-none-any.whl (1.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 68.6 MB/s eta 0:00:00
Requirement already satisfied: decorator in /usr/local/lib/python3.10/dist-packages (from ipython->s
Requirement already satisfied: pickleshare in /usr/local/lib/python3.10/dist-packages (from ipython-
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.10/dist-packages (from ipyth
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /usr/local/lib/python
Requirement already satisfied: pygments in /usr/local/lib/python3.10/dist-packages (from ipython->sa
Requirement already satisfied: backcall in /usr/local/lib/python3.10/dist-packages (from ipython->sa
Requirement already satisfied: matplotlib-inline in /usr/local/lib/python3.10/dist-packages (from ip
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.10/dist-packages (from ipython-
Collecting antlr4-python3-runtime==4.9.* (from omegaconf->salesforce-lavis)
  Downloading antlr4-python3-runtime-4.9.3.tar.gz (117 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 117.0/117.0 kB 18.6 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: kaggle in /usr/local/lib/python3.10/dist-packages (from opendatasets-
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from opendatasets->
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas-
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from panda
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plot
Collecting cfgv>=2.0.0 (from pre-commit->salesforce-lavis)
  Downloading cfgv-3.4.0-py2.py3-none-any.whl (7.2 kB)
Collecting identify>=1.0.0 (from pre-commit->salesforce-lavis)
  Downloading identify-2.5.36-py2.py3-none-any.whl (98 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 99.0/99.0 kB 13.5 MB/s eta 0:00:00
Collecting nodeenv>=0.11.1 (from pre-commit->salesforce-lavis)
  Downloading nodeenv-1.8.0-py2.py3-none-any.whl (22 kB)
Collecting virtualenv>=20.10.0 (from pre-commit->salesforce-lavis)
  Downloading virtualenv-20.26.2-py3-none-any.whl (3.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.9/3.9 MB 85.1 MB/s eta 0:00:00
Requirement already satisfied: matplotlib>=2.1.0 in /usr/local/lib/python3.10/dist-packages (from py
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from scikit-
Requirement already satisfied: pillow!=7.1.0,!=7.1.1,!=8.3.0,>=6.1.0 in /usr/local/lib/python3.10/di
Requirement already satisfied: imageio>=2.4.1 in /usr/local/lib/python3.10/dist-packages (from sciki
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from sc
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: thinc<8.3.0,>=8.2.2 in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: weasel<0.4.0,>=0.1.0 in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: typer<0.10.0,>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.10/dis
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: altair<6,>=4.0 in /usr/local/lib/python3.10/dist-packages (from strea
Requirement already satisfied: blinker<2,>=1.0.0 in /usr/lib/python3/dist-packages (from streamlit->
Requirement already satisfied: cachetools<6,>=4.0 in /usr/local/lib/python3.10/dist-packages (from s
Requirement already satisfied: protobuf<5,>=3.20 in /usr/local/lib/python3.10/dist-packages (from st
Requirement already satisfied: pyarrow>=7.0 in /usr/local/lib/python3.10/dist-packages (from streaml
Requirement already satisfied: rich<14,>=10.14.0 in /usr/local/lib/python3.10/dist-packages (from st
Requirement already satisfied: toml<2,>=0.10.1 in /usr/local/lib/python3.10/dist-packages (from stre
Collecting gitpython!=3.1.19,<4,>=3.0.7 (from streamlit->salesforce-lavis)
  Downloading GitPython-3.1.43-py3-none-any.whl (207 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.3/207.3 kB 25.2 MB/s eta 0:00:00
Collecting pydeck<1,>=0.8.0b4 (from streamlit->salesforce-lavis)
  Downloading pydeck-0.9.1-py2.py3-none-any.whl (6.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.9/6.9 MB 119.0 MB/s eta 0:00:00
Requirement already satisfied: tornado<7,>=6.0.3 in /usr/local/lib/python3.10/dist-packages (from st
Collecting watchdog>=2.1.5 (from streamlit->salesforce-lavis)
  Downloading watchdog-4.0.0-py3-none-manylinux2014_x86_64.whl (82 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 83.0/83.0 kB 13.6 MB/s eta 0:00:00
Collecting braceexpand (from webdataset->salesforce-lavis)
  Downloading braceexpand-0.1.7-py2.py3-none-any.whl (5.9 kB)
Requirement already satisfied: entrypoints in /usr/local/lib/python3.10/dist-packages (from altair<6
Requirement already satisfied: jsonschema>=3.0 in /usr/local/lib/python3.10/dist-packages (from alta
Requirement already satisfied: toolz in /usr/local/lib/python3.10/dist-packages (from altair<6,>=4.0
Collecting gitdb<5,>=4.0.1 (from gitpython!=3.1.19,<4,>=3.0.7->streamlit->salesforce-lavis)
  Downloading gitdb-4.0.11-py3-none-any.whl (62 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.7/62.7 kB 10.0 MB/s eta 0:00:00
Requirement already satisfied: parso<0.9.0,>=0.8.3 in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: language-data>=1.2 in /usr/local/lib/python3.10/dist-packages (from l
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from mat
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplot
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from ma
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from ma
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from mat
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.10/dist-packages (from pexp
Requirement already satisfied: annotated-types>=0.4.0 in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: pydantic-core==2.18.2 in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinj
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-date
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from request
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from r
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from r
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: blis<0.8.0,>=0.7.8 in /usr/local/lib/python3.10/dist-packages (from t
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.10/dist-packages (
Collecting distlib<1,>=0.3.7 (from virtualenv>=20.10.0->pre-commit->salesforce-lavis)
  Downloading distlib-0.3.8-py2.py3-none-any.whl (468 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 468.9/468.9 kB 51.5 MB/s eta 0:00:00
Requirement already satisfied: platformdirs<5,>=3.9.1 in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: cloudpathlib<0.17.0,>=0.7.0 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: python-slugify in /usr/local/lib/python3.10/dist-packages (from kaggl
```

```
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from kaggle->opend
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->
Collecting smmap<6,>=3.0.1 (from gitdb<5,>=4.0.1->gitpython!=3.1.19,<4,>=3.0.7->streamlit->salesforc
  Downloading smmap-5.0.1-py3-none-any.whl (24 kB)
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.10/dist-packages (from jsonsc
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/python3.10/dis
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/dist-packages (from jsons
Requirement already satisfied: marisa-trie>=0.7.7 in /usr/local/lib/python3.10/dist-packages (from l
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from bleach-
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.10/dist-packages (from
Building wheels for collected packages: fairscale, contexttimer, iopath, antlr4-python3-runtime
  Building wheel for fairscale (pyproject.toml) ... done
  Created wheel for fairscale: filename=fairscale-0.4.4-py3-none-any.whl size=292833 sha256=166c65ec
  Stored in directory: /root/.cache/pip/wheels/08/58/6f/56c57fa8315eb0bcf0287b580c850845be5f116359b8
  Building wheel for contexttimer (setup.py) ... done
  Created wheel for contexttimer: filename=contexttimer-0.3.3-py3-none-any.whl size=5804 sha256=6252
  Stored in directory: /root/.cache/pip/wheels/72/1c/da/cfd97201d88ccce214427fa84a5caeb91fef7c5a1b4c
  Building wheel for iopath (setup.py) ... done
  Created wheel for iopath: filename=iopath-0.1.10-py3-none-any.whl size=31532 sha256=60a4d02bf63f04
  Stored in directory: /root/.cache/pip/wheels/9a/a3/b6/ac0fcd1b4ed5cfeb3db92e6a0e476cfd48ed0df92b91
  Building wheel for antlr4-python3-runtime (setup.py) ... done
  Created wheel for antlr4-python3-runtime: filename=antlr4_python3_runtime-4.9.3-py3-none-any.whl s
  Stored in directory: /root/.cache/pip/wheels/12/93/dd/1f6a127edc45659556564c5730f6d4e300888f4bca2d
Successfully built fairscale contexttimer iopath antlr4-python3-runtime
Installing collected packages: tokenizers, distlib, contexttimer, braceexpand, antlr4-python3-runtim
  Attempting uninstall: tokenizers
    Found existing installation: tokenizers 0.19.1
    Uninstalling tokenizers-0.19.1:
      Successfully uninstalled tokenizers-0.19.1
  Attempting uninstall: opencv-python-headless
    Found existing installation: opencv-python-headless 4.9.0.80
    Uninstalling opencv-python-headless-4.9.0.80:
      Successfully uninstalled opencv-python-headless-4.9.0.80
  Attempting uninstall: transformers
    Found existing installation: transformers 4.40.2
    Uninstalling transformers-4.40.2:
      Successfully uninstalled transformers-4.40.2
Successfully installed antlr4-python3-runtime-4.9.3 braceexpand-0.1.7 cfgv-3.4.0 contexttimer-0.3.3
```

```
import torch
from lavis.models import load_model_and_preprocess
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model, vis_processors, _ = load_model_and_preprocess(name="blip_caption", model_type="base_coco", is_eval=True, device=device)
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or dataset
  warnings.warn(
vocab.txt: 100%                                   232k/232k [00:00<00:00, 9.18MB/s]
tokenizer_config.json: 100%                       48.0/48.0 [00:00<00:00, 3.31kB/s]
config.json: 100%                                 570/570 [00:00<00:00, 38.4kB/s]
100%|██████████| 2.50G/2.50G [01:53<00:00, 23.6MB/s]
```

```
with autocast(device):
  textprompt = str(input("Enter your prompt: "))

  image = pipe(textprompt, guidance_scale=8.5).images[0]

  imgplot = plt.imshow(image)
```

```
Enter your prompt: A girl talking on a phone
100%                                              50/50 [00:08<00:00,  6.15it/s]
```

SnapScore using Lavis Model

We have generated captions using Salesforce-Lavis for three examples and compared SnapScore for those obtained from our LSTM model

eg1

```
from google.colab import files
uploaded = files.upload()
```

```
Choose Files No file chosen  Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving Apple and banana.png to Apple and banana.png
```

```
# Import necessary libraries
from PIL import Image
import torch

# Load the image
image_path = "Apple and banana.png"
image = Image.open(image_path)

# Convert RGBA to RGB if necessary
if image.mode == 'RGBA':
    image = image.convert('RGB')

# Assuming 'vis_processors' and 'model' are defined and 'device' is set
# Preprocess the image
image = vis_processors["eval"](image).unsqueeze(0).to(device)

# Generate caption
caption = model.generate({"image": image})

# Print the generated caption
print(caption)
```

```
['a bunch of bananas hanging from a tree']
```

BLEU

```
import nltk
nltk.download('punkt')

from nltk.translate.bleu_score import sentence_bleu, SmoothingFunction
from nltk.tokenize import word_tokenize
# Example sentences
reference = "a bunch of bananas and fruits hanging on a tree"
candidate = "a bunch of bananas hanging from a tree"

# Tokenize the sentences
ref_tokens = [word_tokenize(reference)]
cand_tokens = word_tokenize(candidate)

# Smooth BLEU score calculation
smoothie = SmoothingFunction().method4  # Adjust the smoothing function as needed
scoreBLEU = sentence_bleu(ref_tokens, cand_tokens, smoothing_function=smoothie)
print(f"Smoothed BLEU Score: {scoreBLEU:.4f}")
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
Smoothed BLEU Score: 0.3328
```

ROUGE

```
!pip install rouge

from rouge import Rouge

# Example sentences
reference = "a bunch of bananas and fruits hanging on a tree"
candidate = "a bunch of bananas hanging from a tree"

# Initialize Rouge
rouge = Rouge()

# Calculate scores
scores = rouge.get_scores(candidate, reference)
print(scores)
```

```
Collecting rouge
  Downloading rouge-1.0.1-py3-none-any.whl (13 kB)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from rouge) (1.16.0)
Installing collected packages: rouge
Successfully installed rouge-1.0.1
[{'rouge-1': {'r': 0.6666666666666666, 'p': 0.8571428571428571, 'f': 0.7499999950781251}, 'rouge-2': {'r': 0.4444444444444444, 'p': 0.5714285714285714, 'f
```

```
from rouge import Rouge

# Example sentences
reference = "a bunch of bananas and fruits hanging on a tree"
candidate = "a bunch of bananas hanging from a tree"

# Initialize Rouge
rouge = Rouge()

# Calculate scores
scores = rouge.get_scores(candidate, reference)[0]  # Access the first (and only) set of scores

# Define a smoothing function
def smooth_scores(scores, alpha=0.1):
    for key in scores:
        scores[key]['p'] = (scores[key]['p'] + alpha) / (1 + alpha)
        scores[key]['r'] = (scores[key]['r'] + alpha) / (1 + alpha)
        scores[key]['f'] = (scores[key]['f'] + alpha) / (1 + alpha)
    return scores

# Apply smoothing
smoothed_scores = smooth_scores(scores)

print(smoothed_scores)

average_f1 = (scores['rouge-1']['f'] + scores['rouge-2']['f'] + scores['rouge-1']['f']) / 3
print("Average ROUGE F1 Score:", average_f1)
```

```
{'rouge-1': {'r': 0.6969696969696969, 'p': 0.87012987012987, 'f': 0.7727272682528409}, 'rouge-2': {'r': 0.49494949494949486, 'p': 0.6103896103896103, 'f':
Average ROUGE F1 Score: 0.6969696924952652
```

METEOR

```
import nltk

# Download the WordNet data
nltk.download('wordnet')

# After downloading, you can continue with the METEOR score calculation
from nltk.translate.meteor_score import meteor_score

# Example sentences
reference = "a bunch of bananas and fruits hanging on a tree"
candidate = "a bunch of bananas hanging from a tree"

# Tokenize both the reference and hypothesis
reference_tokens = reference.split()  # Tokenize reference
hypothesis_tokens = candidate.split()  # Tokenize hypothesis

# Calculate METEOR score
METEORscore = meteor_score([reference_tokens], hypothesis_tokens)
print("METEOR Score:", METEORscore)
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
METEOR Score: 0.6861724281549355
```

cosine

```
!pip install scikit-learn

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# Example texts
text1 = "a bunch of bananas and fruits hanging on a tree"
text2 = "a bunch of bananas hanging from a tree"

# Initialize a vectorizer
vectorizer = TfidfVectorizer()

# Vectorize the texts
tfidf = vectorizer.fit_transform([text1, text2])

# Calculate cosine similarity
cosine_sim = cosine_similarity(tfidf[0:1], tfidf[1:2])[0][0]

print("Cosine Similarity:", cosine_sim)
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.25.2)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.4)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
Cosine Similarity: 0.5727393584196196
```

snapscore

```
weights = {
    'BLEU': 0.20,    # Lower weight due to focus on exact n-gram matches
    'ROUGE': 0.20,   # Similar reason as BLEU
    'METEOR': 0.25,  # Higher weight as it accounts for synonyms and structure
    'cosine': 0.35   # Highest weight, focusing on semantic similarity
}


scores = {
    'BLEU': 0.3328,      # Example BLEU score
    'ROUGE': 0.6969696924952652,    # Example ROUGE score
    'METEOR': 0.6861724281549355,   # Example METEOR score
    'cosine': 0.5727393584196196    # Example cosine similarity score
}



# Calculate the SNAP Score as a weighted sum of the scores
snap_score = sum(weights[metric] * scores[metric] for metric in weights)

print("SNAP Score:", snap_score)
```

```
SNAP Score: 0.5779558209846538
```

eg2

```
from google.colab import files
uploaded = files.upload()
```

```
Choose Files No file chosen  Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving doctor.png to doctor.png
```

BLEU

```
# Import necessary libraries
from PIL import Image
import torch

# Load the image
image_path = "doctor.png"
image = Image.open(image_path)

# Convert RGBA to RGB if necessary
if image.mode == 'RGBA':
    image = image.convert('RGB')

# Assuming 'vis_processors' and 'model' are defined and 'device' is set
# Preprocess the image
image = vis_processors["eval"](image).unsqueeze(0).to(device)

# Generate caption
caption = model.generate({"image": image})

# Print the generated caption
print(caption)
```

```
['a man wearing a blue lab coat and a stethoscope']
```

```
import nltk
nltk.download('punkt')

from nltk.translate.bleu_score import sentence_bleu, SmoothingFunction
from nltk.tokenize import word_tokenize
# Example sentences
reference = "a man wearing a suit and tie standing with a stethoscope"
candidate = "a man wearing a blue lab coat and a stethoscope"

# Tokenize the sentences
ref_tokens = [word_tokenize(reference)]
cand_tokens = word_tokenize(candidate)

# Smooth BLEU score calculation
smoothie = SmoothingFunction().method4  # Adjust the smoothing function as needed
scoreBLEU = sentence_bleu(ref_tokens, cand_tokens, smoothing_function=smoothie)
print(f"Smoothed BLEU Score: {scoreBLEU:.4f}")
```

```
Smoothed BLEU Score: 0.2938
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

ROUGE

```
from rouge import Rouge

# Example sentences
reference = "a man wearing a suit and tie standing with a stethoscope"
candidate = "a man wearing a blue lab coat and a stethoscope"

# Initialize Rouge
rouge = Rouge()

# Calculate scores
scores = rouge.get_scores(candidate, reference)
print(scores)
```

```
[{'rouge-1': {'r': 0.5555555555555556, 'p': 0.625, 'f': 0.5882352891349482}, 'rouge-2': {'r': 0.4, 'p': 0.4444444444444444, 'f': 0.42105262659279785}, 'ro
```

```
from rouge import Rouge

# Example sentences
reference = "a man wearing a suit and tie standing with a stethoscope"
candidate = "a man wearing a blue lab coat and a stethoscope"

# Initialize Rouge
rouge = Rouge()

# Calculate scores
scores = rouge.get_scores(candidate, reference)[0]  # Access the first (and only) set of scores

# Define a smoothing function
def smooth_scores(scores, alpha=0.1):
    for key in scores:
        scores[key]['p'] = (scores[key]['p'] + alpha) / (1 + alpha)
        scores[key]['r'] = (scores[key]['r'] + alpha) / (1 + alpha)
        scores[key]['f'] = (scores[key]['f'] + alpha) / (1 + alpha)
    return scores

# Apply smoothing
smoothed_scores = smooth_scores(scores)

print(smoothed_scores)

average_f1 = (scores['rouge-1']['f'] + scores['rouge-2']['f'] + scores['rouge-l']['f']) / 3
print("Average ROUGE F1 Score:", average_f1)
```

```
{'rouge-1': {'r': 0.5959595959595959, 'p': 0.6590909090909091, 'f': 0.6256684446681346}, 'rouge-2': {'r': 0.45454545454545453, 'p': 0.49494949494949486, '
Average ROUGE F1 Score: 0.575007031776574
```

METEOR

```
import nltk

# Download the WordNet data
nltk.download('wordnet')

# After downloading, you can continue with the METEOR score calculation
from nltk.translate.meteor_score import meteor_score

# Example sentences
reference = "a man wearing a suit and tie standing with a stethoscope"
candidate = "a man wearing a blue lab coat and a stethoscope"

# Tokenize both the reference and hypothesis
reference_tokens = reference.split()  # Tokenize reference
hypothesis_tokens = candidate.split()  # Tokenize hypothesis

# Calculate METEOR score
METEORscore = meteor_score([reference_tokens], hypothesis_tokens)
print("METEOR Score:", METEORscore)
```

```
METEOR Score: 0.616925669350309
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

cosine

```
!pip install scikit-learn

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# Example texts
text1 = "a man wearing a suit and tie standing with a stethoscope"
text2 = "a man wearing a blue lab coat and a stethoscope"

# Initialize a vectorizer
vectorizer = TfidfVectorizer()

# Vectorize the texts
tfidf = vectorizer.fit_transform([text1, text2])

# Calculate cosine similarity
cosine_sim = cosine_similarity(tfidf[0:1], tfidf[1:2])[0][0]

print("Cosine Similarity:", cosine_sim)
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.25.2)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.4)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
Cosine Similarity: 0.36802320875611494
```

snapscore

```
weights = {
    'BLEU': 0.20,    # Lower weight due to focus on exact n-gram matches
    'ROUGE': 0.20,   # Similar reason as BLEU
    'METEOR': 0.25,  # Higher weight as it accounts for synonyms and structure
    'cosine': 0.35   # Highest weight, focusing on semantic similarity
}


scores = {
    'BLEU': 0.2938,      # Example BLEU score
    'ROUGE': 0.575007031776574,    # Example ROUGE score
    'METEOR': 0.616925669350309,   # Example METEOR score
    'cosine': 0.36802320875611494    # Example cosine similarity score
}



# Calculate the SNAP Score as a weighted sum of the scores
snap_score = sum(weights[metric] * scores[metric] for metric in weights)

print("SNAP Score:", snap_score)
```

```
SNAP Score: 0.4568009467575323
```

snap score for animal eating grass, detailed, 8k

```
from google.colab import files
uploaded = files.upload()
```

```
Choose Files No file chosen  Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving anaimal_grass.png to anaimal_grass.png
```

BLEU

```
# Import necessary libraries
from PIL import Image
import torch

# Load the image
image_path = "anaimal_grass.png"
image = Image.open(image_path)

# Convert RGBA to RGB if necessary
if image.mode == 'RGBA':
    image = image.convert('RGB')

# Assuming 'vis_processors' and 'model' are defined and 'device' is set
# Preprocess the image
image = vis_processors["eval"](image).unsqueeze(0).to(device)

# Generate caption
caption = model.generate({"image": image})

# Print the generated caption
print(caption)
```

```
['a close up of a dog with its mouth open']
```

```
import nltk
nltk.download('punkt')

from nltk.translate.bleu_score import sentence_bleu, SmoothingFunction
from nltk.tokenize import word_tokenize
# Example sentences
reference = "animal standing in the grass with its mouth open"
candidate = "a close up of a dog with its mouth open"

# Tokenize the sentences
ref_tokens = [word_tokenize(reference)]
cand_tokens = word_tokenize(candidate)

# Smooth BLEU score calculation
smoothie = SmoothingFunction().method4  # Adjust the smoothing function as needed
scoreBLEU = sentence_bleu(ref_tokens, cand_tokens, smoothing_function=smoothie)
print(f"Smoothed BLEU Score: {scoreBLEU:.4f}")
```

```
Smoothed BLEU Score: 0.2627
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

ROUGE

```
from rouge import Rouge

# Example sentences
reference = "animal standing in the grass with its mouth open"
candidate = "a close up of a dog with its mouth open"

# Initialize Rouge
rouge = Rouge()

# Calculate scores
scores = rouge.get_scores(candidate, reference)
print(scores)
```

```
[{'rouge-1': {'r': 0.4444444444444444, 'p': 0.4444444444444444, 'f': 0.44444443944444445}, 'rouge-2': {'r': 0.375, 'p': 0.3333333333333333, 'f': 0.3529411
```

```
from rouge import Rouge

# Example sentences
reference = "animal standing in the grass with its mouth open"
candidate = "a close up of a dog with its mouth open"

# Initialize Rouge
rouge = Rouge()

# Calculate scores
scores = rouge.get_scores(candidate, reference)[0]  # Access the first (and only) set of scores

# Define a smoothing function
def smooth_scores(scores, alpha=0.1):
    for key in scores:
        scores[key]['p'] = (scores[key]['p'] + alpha) / (1 + alpha)
        scores[key]['r'] = (scores[key]['r'] + alpha) / (1 + alpha)
        scores[key]['f'] = (scores[key]['f'] + alpha) / (1 + alpha)
    return scores

# Apply smoothing
smoothed_scores = smooth_scores(scores)

print(smoothed_scores)

average_f1 = (scores['rouge-1']['f'] + scores['rouge-2']['f'] + scores['rouge-l']['f']) / 3
print("Average ROUGE F1 Score:", average_f1)
```

```
{'rouge-1': {'r': 0.49494949494949486, 'p': 0.49494949494949486, 'f': 0.49494949040404035}, 'rouge-2': {'r': 0.43181818181818177, 'p': 0.3939393939393939,
Average ROUGE F1 Score: 0.46722122738690247
```

METEOR

```
import nltk

# Download the WordNet data
nltk.download('wordnet')

# After downloading, you can continue with the METEOR score calculation
from nltk.translate.meteor_score import meteor_score
```

cosine

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# Example texts
text1 = "animal standing in the grass with its mouth open"
text2 = "a close up of a dog with its mouth open"

# Initialize a vectorizer
vectorizer = TfidfVectorizer()

# Vectorize the texts
tfidf = vectorizer.fit_transform([text1, text2])

# Calculate cosine similarity
cosine_sim = cosine_similarity(tfidf[0:1], tfidf[1:2])[0][0]

print("Cosine Similarity:", cosine_sim)
```

```
Cosine Similarity: 0.31125746752705374
```

snapscore

```
weights = {
    'BLEU': 0.20,    # Lower weight due to focus on exact n-gram matches
    'ROUGE': 0.20,   # Similar reason as BLEU
    'METEOR': 0.25,  # Higher weight as it accounts for synonyms and structure
    'cosine': 0.35   # Highest weight, focusing on semantic similarity
}
```

```
scores = {
    'BLEU': 0.2627,      # Example BLEU score
    'ROUGE': 0.46722122738690247,    # Example ROUGE score
    'METEOR': 0.4361263736263736,   # Example METEOR score
    'cosine': 0.31125746752705374    # Example cosine similarity score
}
```

```
# Calculate the SNAP Score as a weighted sum of the scores
snap_score = sum(weights[metric] * scores[metric] for metric in weights)

print("SNAP Score:", snap_score)
```

```
SNAP Score: 0.3639559525184427
```

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit