Student: Iskander Kushbay

Class: CSE472 – Social Media Mining

Project II, Type 2

Multimodal Counterfactual Data Generation and Evaluation

1. Abstract

As generative models continue to advance, their applications are expanding across various domains, including commercial content creation, data modification, and scenario modeling. These models are categorized based on the type of content they generate—such as videos, text, images, or other formats. One notable application of generative models is in the creation and modification of examples for tasks like training other models. This is where counterfactual data generation proves valuable. This project focuses on exploring methods for generating counterfactual data in both textual and visual formats, developing a comprehensive framework for this purpose, and evaluating its quality.

2. Introduction

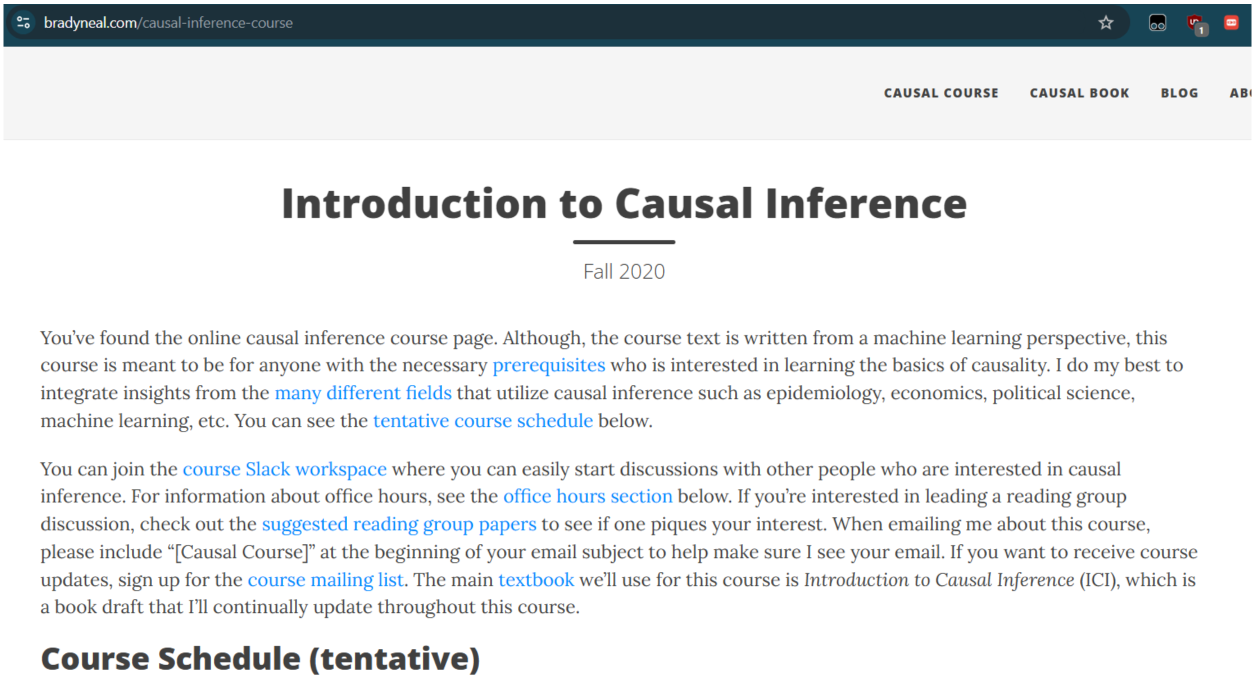
Generative models (for example, GPT) rely on some internal representation of the world. The representation, however, is mechanically uninterpretable - a person cannot look inside the model, look at the collection of weights and make unambiguous conclusions. And yet, as practice shows, the model relies on a chain of thoughts when issuing an answer. Roughly speaking, it is marked by the principle of causality.

Causal inference studies how information is transferred between dependent entities in chains. SCM (structural causal model) is a directed acyclic graph (DAG), which is an interpretation of Causal inference in a system. This project uses the concept of SCM for the initial representation of a prompt and its subsequent modification in order to generate counterfactual data.

One of the goals of this project is to study how well the generation of counterfactual data is scaled. For this purpose, the project adapted models for generating text and images to study how well the framework proposed in the project is revealed at scale. The language model, given a prompt, generates a SCM and, based on it, several counterfactual labels, which are then used to generate counterfactual images.

3. Related Works

The project is based on the concept of the SCM, which is used in many areas, from biology to economics. For this project, the Causal Inference course provided by the project curator was completed.



Introduction to Causal Inference

Fall 2020

You've found the online causal inference course page. Although, the course text is written from a machine learning perspective, this course is meant to be for anyone with the necessary prerequisites who is interested in learning the basics of causality. I do my best to integrate insights from the many different fields that utilize causal inference such as epidemiology, economics, political science, machine learning, etc. You can see the tentative course schedule below.

You can join the course Slack workspace where you can easily start discussions with other people who are interested in causal inference. For information about office hours, see the office hours section below. If you're interested in leading a reading group discussion, check out the suggested reading group papers to see if one piques your interest. When emailing me about this course, please include "[Causal Course]" at the beginning of your email subject to help make sure I see your email. If you want to receive course updates, sign up for the course mailing list. The main textbook we'll use for this course is *Introduction to Causal Inference* (ICI), which is a book draft that I'll continually update throughout this course.

Course Schedule (tentative)

The technical implementation of the project was based on two studies (COCO-Counterfactuals: Automatically Constructed Counterfactual Examples for Image-Text Pairs and Benchmarking Counterfactual Image Generation). The first of them focuses on the study of a benchmark using the CCM on different architectures in changing existing datasets to generate counterfactual data, which is very close to the topic of the project. The other study shows how the existing COCO dataset can be modified to create counterfactual data, and generative model metrics are used for this purpose. This project takes a little from everywhere and compiles existing knowledge

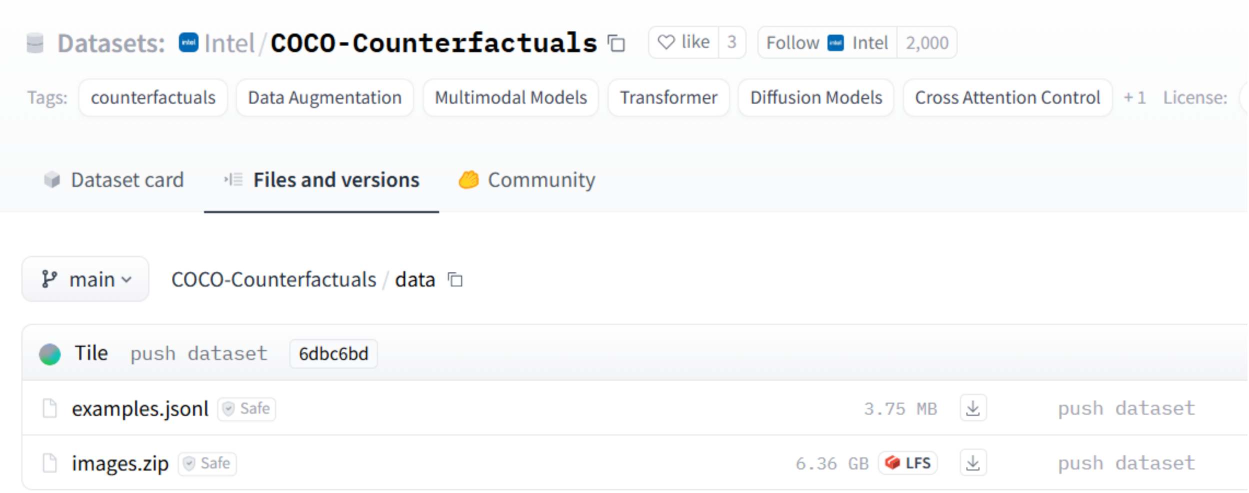
4. Model Description

For this project, the idea of using one model to generate counterfactual prompts and another to generate images for this same prompt was adapted. The COCO dataset was used as a basis, from which some of the labels were taken, and counterfactual data were generated based on them.

Llama 3.1 Turbo 70B was chosen as the generative model for creating labels. Stable Diffusion ver. 1.4 was used as the model for creating images. After that, metrics are evaluated using CLIP and various libraries.

5. Experiment (or Methodic)

1) To begin with, the COCO dataset was selected and downloaded, from which the first 10 objects were selected in order to generate the required data based on them.

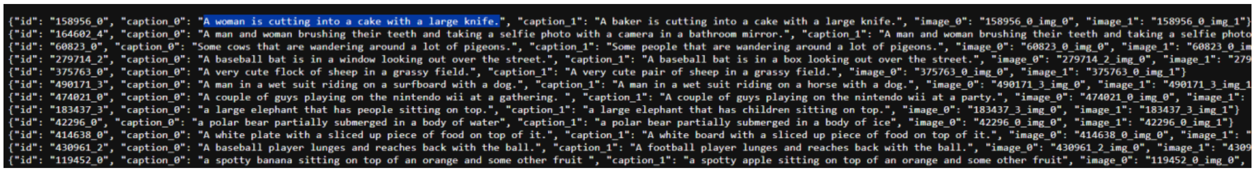


Datasets: Intel / COCO-Counterfactuals

COCO-Counterfactuals / data

Tile push dataset 6dbc6bd

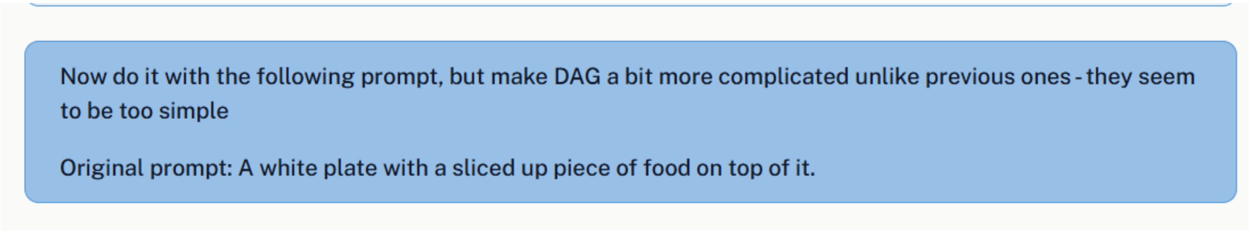
examples.jsonl Safe	3.75 MB		push dataset
images.zip Safe	6.36 GB	LFS	push dataset



```
{"id": "158956_0", "caption_0": "A woman is cutting into a cake with a large knife.", "caption_1": "A baker is cutting into a cake with a large knife.", "image_0": "158956_0_img_0", "image_1": "158956_0_img_1"}
{"id": "164602_4", "caption_0": "A man and woman brushing their teeth and taking a selfie photo with a camera in a bathroom mirror.", "caption_1": "A man and woman brushing their teeth and taking a selfie photo
{"id": "60823_0", "caption_0": "Some cows that are wandering around a lot of pigeons.", "caption_1": "Some people that are wandering around a lot of pigeons.", "image_0": "60823_0_img_0", "image_1": "60823_0_im
{"id": "279714_2", "caption_0": "A baseball bat is in a window looking out over the street.", "caption_1": "A baseball bat is in a box looking out over the street.", "image_0": "279714_2_img_0", "image_1": "279
{"id": "375763_0", "caption_0": "A very cute flock of sheep in a grassy field.", "caption_1": "A very cute pair of sheep in a grassy field.", "image_0": "375763_0_img_0", "image_1": "375763_0_img_1"}
{"id": "490171_3", "caption_0": "A man in a wet suit riding on a surfboard with a dog.", "caption_1": "A man in a wet suit riding on a horse with a dog.", "image_0": "490171_3_img_0", "image_1": "490171_3_img_1
{"id": "474021_0", "caption_0": "A couple of guys playing on the nintendo wii at a gathering. ", "caption_1": "A couple of guys playing on the nintendo wii at a party.", "image_0": "474021_0_img_0", "image_1":
{"id": "183437_3", "caption_0": "a large elephant that has people sitting on top.", "caption_1": "a large elephant that has children sitting on top.", "image_0": "183437_3_img_0", "image_1": "183437_3_img_1"}
{"id": "42296_0", "caption_0": "a polar bear partially submerged in a body of water", "caption_1": "a polar bear partially submerged in a body of ice", "image_0": "42296_0_img_0", "image_1": "42296_0_img_1"}
{"id": "414638_0", "caption_0": "A white plate with a sliced up piece of food on top of it.", "caption_1": "A white board with a sliced up piece of food on top of it.", "image_0": "414638_0_img_0", "image_1": "
{"id": "430961_2", "caption_0": "A baseball player lunges and reaches back with the ball.", "caption_1": "A football player lunges and reaches back with the ball.", "image_0": "430961_2_img_0", "image_1": "4309
{"id": "119452_0", "caption_0": "a spotty banana sitting on top of an orange and some other fruit ", "caption_1": "a spotty apple sitting on top of an orange and some other fruit", "image_0": "119452_0_img_0",
```

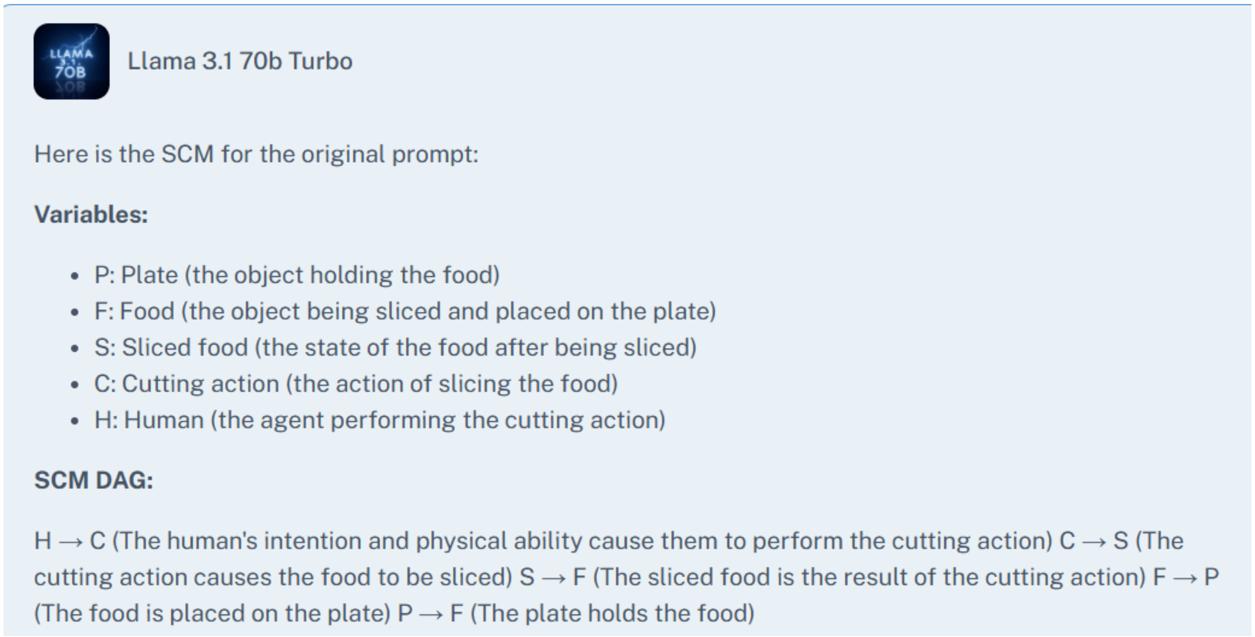
The structure of this dataset was also taken as a basis to associate the resulting labels with the generated images.

2) Next, a language model (in our case, LLama 3.1B Turbo) was used to generate the SCM based on the original prompt. In order to somehow standardize the responses, certain prompts were entered, and after that, the model began to generate the SCM along with the modified prompt and wrap it all in a convenient json.



Now do it with the following prompt, but make DAG a bit more complicated unlike previous ones - they seem to be too simple

Original prompt: A white plate with a sliced up piece of food on top of it.



Llama 3.1 70b Turbo

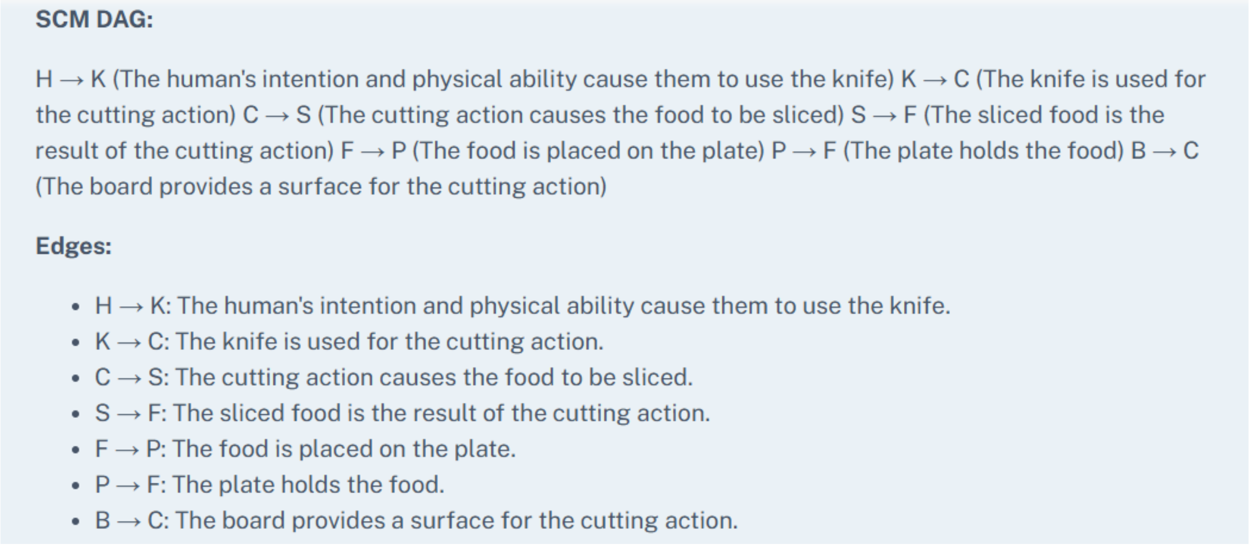
Here is the SCM for the original prompt:

Variables:

- P: Plate (the object holding the food)
- F: Food (the object being sliced and placed on the plate)
- S: Sliced food (the state of the food after being sliced)
- C: Cutting action (the action of slicing the food)
- H: Human (the agent performing the cutting action)

SCM DAG:

H → C (The human's intention and physical ability cause them to perform the cutting action) C → S (The cutting action causes the food to be sliced) S → F (The sliced food is the result of the cutting action) F → P (The food is placed on the plate) P → F (The plate holds the food)



SCM DAG:

H → K (The human's intention and physical ability cause them to use the knife) K → C (The knife is used for the cutting action) C → S (The cutting action causes the food to be sliced) S → F (The sliced food is the result of the cutting action) F → P (The food is placed on the plate) P → F (The plate holds the food) B → C (The board provides a surface for the cutting action)

Edges:

- H → K: The human's intention and physical ability cause them to use the knife.
- K → C: The knife is used for the cutting action.
- C → S: The cutting action causes the food to be sliced.
- S → F: The sliced food is the result of the cutting action.
- F → P: The food is placed on the plate.
- P → F: The plate holds the food.
- B → C: The board provides a surface for the cutting action.

Example of an output json:

```
json_entry_2 = {
    "original_prompt": "A white plate with a sliced up piece of food on top of it.",
    "counterfactual_prompt": "A white plate with a sliced up piece of food on top of it, using a fork to cut.",
    "original_image": "",
    "counterfactual_image": "",
    "SCM": [
        "H → K: The human's intention and physical ability cause them to
use the knife.",
        "K → C: The knife is used for the cutting action.",
        "C → S: The cutting action causes the food to be sliced.",
        "S → F: The sliced food is the result of the cutting action.",
        "F → P: The food is placed on the plate.",
        "P → F: The plate holds the food.",
        "B → C: The board provides a surface for the cutting action."
    ],
    "SCM_aug": [
        "H → Fr (The human's intention and physical ability cause them to
use the fork)",
        "Fr → C (The fork is used for the cutting action)",
        "C → S (The cutting action causes the food to be sliced)",
        "S → F (The sliced food is the result of the cutting action)",
        "F → P (The food is placed on the plate)",
        "P → F (The plate holds the food)",
        "B → C (The board provides a surface for the cutting action)"
    ]
}
```

For each label, 5 counterfactual examples were generated, resulting in a total of 50 labels.

3) The next step was, in fact, generating the images themselves based on the modified prompts. For this, the Stable Diffusion model (1.4) was chosen. Due to the lack of local capacities needed for this, Google Collab was used, so when trying to run the code locally, it should be configured for the local system.
   The generations incorporated two practices from papers – AAP (Abduction, Action, Prediction) and Prompt-to-Prompt generation. The former server to incorporating SCM into the generation process, while he latest is a google research framework for creating images that are closely similar to an original image though attention mask transplantation,

A total of 50 images were generated based on the prompts, and these 50 images can be divided into 10 groups of 5. A file “images_generation.py” (or .ipynb) contains the code for it. The images can be found on a google drive link provided at the end of the document.
All images were saved in the folder and a .json files corresponding to each group were generated as well. Later those json files were assembled into 1 big json “counterfactual_dataset.json”

An example entry:

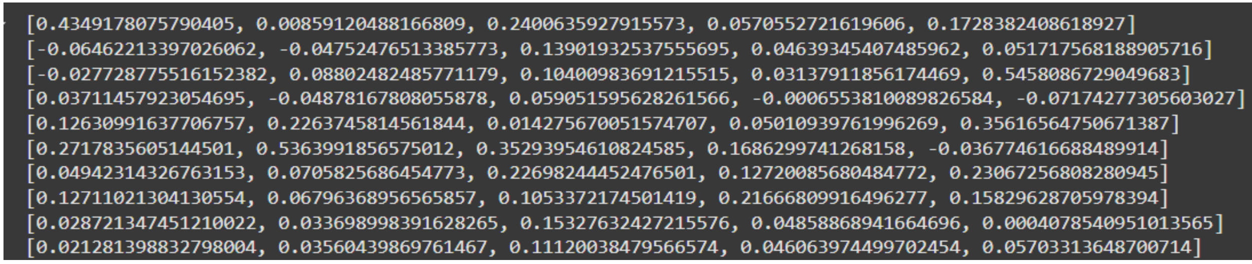
```
{
        "id": 1,
        "original_prompt": "A woman is cutting into a cake with a large
knife.",
        "counterfactual_prompt": "A man is cutting into a cake with a large
knife.",
        "original_image": "original_1.png",
        "counterfactual_image": "counterfactual_1.png",
        "SCM": [
            "W \u2192 A: The woman's intention and physical ability cause
the action of cutting into the cake.",
            "W \u2192 K: The woman's agency and control enable her to hold
the knife.",
            "K \u2192 A: The knife is a necessary tool for the action of
cutting into the cake.",
            "C \u2192 A: The cake's presence and properties (e.g., its texture,
structure) affect the action of cutting into it."
        ],
        "SCM_aug": [
            "M \u2192 A (The man is the cause of the action)",
            "M \u2192 K (The man is holding the knife)",
            "K \u2192 A (The knife is used for the action)",
            "C \u2192 A (The cake is the object being acted upon)"
        ]
    }
```

4) Finally, the quality of the generated label-image pairs should be measured. The file “metric_calculation.py” (or .ipynb) is responsible for this part. Four existing standards were chosen as metrics:
    1) CLIP directional similarity

$$\textbf{CLIP}_{dir} = \frac{(E_T(C_c) - E_T(C_o)) \cdot (E_I(I_c^s) - E_I(I_o^s))}{||E_T(C_c) - E_T(C_o)||\ ||E_I(I_c^s) - E_I(I_o^s)||}$$

This metric shows how well the counterfactual label-image pair matches the original prompt and image. The data is tokenized and transformed into the internal space of the CLIP model, from where the cosine distance between the resulting vectors is calculated.

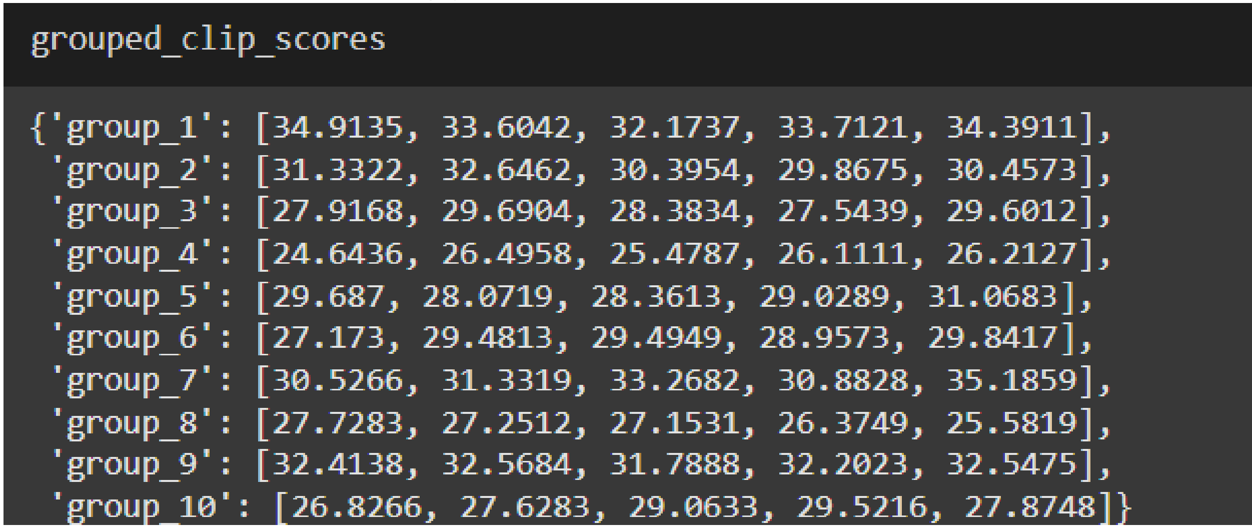
Calculated values for each group and entries:



```
[0.4349178075790405, 0.00859120488166809, 0.2400635927915573, 0.0570552721619606, 0.1728382408618927]
[-0.06462213397026062, -0.04752476513385773, 0.13901932537555695, 0.04639345407485962, 0.051717568188905716]
[-0.027728775516152382, 0.08802482485771179, 0.10400983691215515, 0.03137911856174469, 0.5458086729049683]
[0.03711457923054695, -0.04878167808055878, 0.059051595628261566, -0.0006553810089826584, -0.07174277305603027]
[0.12630991637706757, 0.2263745814561844, 0.014275670051574707, 0.05010939761996269, 0.35616564750671387]
[0.2717835605144501, 0.5363991856575012, 0.35293954610824585, 0.1686299741268158, -0.036774616688489914]
[0.04942314326763153, 0.0705825686454773, 0.22698244452476501, 0.12720085680484772, 0.23067256808280945]
[0.12711021304130554, 0.06796368956565857, 0.1053372174501419, 0.21666809916496277, 0.15829628705978394]
[0.028721347451210022, 0.033698998391628265, 0.15327632427215576, 0.04858868941664696, 0.0004078540951013565]
[0.021281398832798004, 0.03560439869761467, 0.11120038479566574, 0.046063974499702454, 0.05703313648700714]
```

We see that the data fluctuates around 0-0.4, which means that there are discrepancies between the prompt and the image. But they are not so critical as to produce the exact opposite result. In some cases, the direction match is good enough to indicate a small change in generation. On average, this metric indicates that the framework can show itself well. Data was stored in group_scores.json

2) CLIP score
Another metric, this time showing how well the label-picture data correlates for each individual generation. In general, it was chosen to assess the quality of generation, but in my opinion it is not very representative, since it strongly depends on which models are used.



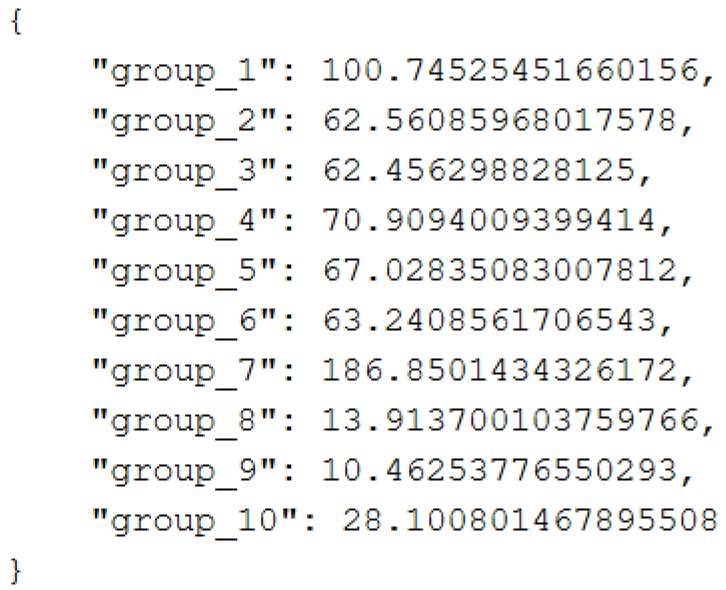
```
grouped_clip_scores
```

```
{'group_1': [34.9135, 33.6042, 32.1737, 33.7121, 34.3911],
 'group_2': [31.3322, 32.6462, 30.3954, 29.8675, 30.4573],
 'group_3': [27.9168, 29.6904, 28.3834, 27.5439, 29.6012],
 'group_4': [24.6436, 26.4958, 25.4787, 26.1111, 26.2127],
 'group_5': [29.687, 28.0719, 28.3613, 29.0289, 31.0683],
 'group_6': [27.173, 29.4813, 29.4949, 28.9573, 29.8417],
 'group_7': [30.5266, 31.3319, 33.2682, 30.8828, 35.1859],
 'group_8': [27.7283, 27.2512, 27.1531, 26.3749, 25.5819],
 'group_9': [32.4138, 32.5684, 31.7888, 32.2023, 32.5475],
 'group_10': [26.8266, 27.6283, 29.0633, 29.5216, 27.8748]}
```

The values fluctuate around 20-30, which indicates a threshold of good generation.
Data was stored in group_clip_scores.json

3) Realism (measured by FID – Frechet Inception Distance)

A metric taken from one of the articles. It shows how similar the original image generation is to the modified one.



```
{
    "group_1": 100.74525451660156,
    "group_2": 62.56085968017578,
    "group_3": 62.456298828125,
    "group_4": 70.9094009399414,
    "group_5": 67.02835083007812,
    "group_6": 63.2408561706543,
    "group_7": 186.850143432617 2,
    "group_8": 13.913700103759766,
    "group_9": 10.46253776550293,
    "group_10": 28.100801467895508
}
```

Measured by groups, because it measures differences in datasets. Quality indicates average to very good matches with the original datasets. Data was stored in grouped_fid_scores.json

4) Minimality (CLD – Counterfactual Latent Divergence)
   The metric shows how well the changed prompt changes only what needs to be changed. Two versions were measured - l2 and cosine distance between generations. In general - good agreement between generations. Data was stored in cosine_cld_grouped.json and l2_cld_grouped.json

6. Future Works

Overall, working on the framework has shown that generating data at scale is possible and has a place to be. Counterfactual data shows that generative AI has the potential to produce modified data of fairly good quality, although it depends on many variables, such as the quality of the models, the work of the prompt, and the availability of sufficient computing power (which I lacked). In the future, it may be possible to add a counterfactual measurement of existing datasets and work on different combinations of models.

7. References

google drive link for images: https://drive.google.com/drive/folders/1qykrk_KI65LGhX79t8R0zzvq9zbCWdWQ?usp=sharing
Benchmarking paper: https://arxiv.org/pdf/2309.14356
COCO Counterfactuals paper: https://arxiv.org/pdf/2403.20287
Huggingface metrics implementation: https://huggingface.co/docs/diffusers/conceptual/evaluation#class-conditioned-image-generation
Causal Inference Course: https://www.bradyneal.com/causal-inference-course
Prompt-to-prompt repo: https://github.com/google/prompt-to-prompt