

Homework 9: Final project preparation

Due April 22nd at 10pm

1 Literature Review

To prepare for the final project, you should do some background reading. Read each of the papers listed below, and take notes so that you can refer back to them as you develop your project.

- [Ted Underwood's response to the Stochastic Parrots paper](#)
- [Blodgett et al. \(2021\)](#)
- [Sheng et al. \(2021\)](#)
- [Zhou, Ethayarajh, & Jurafsky \(2021\)](#)

(If you have another paper that is relevant to your topic, you can substitute it for the last paper listed above.)

For this part of the homework, you should submit a 1 paragraph summary of each paper.

2 Assembling on Your Probe Task

Your main task in this assignment is to make progress on your probe task. I would like you to submit a preliminary dataset for your task.

You must construct at least 80 items for your task.

In addition, your dataset must have **multiple conditions**. You should have parallel items across conditions: that is, if you have two conditions, you should have two versions of each item that are identical except for the factor of interest that you are studying.

You should decide what kind of prompting paradigm you will use. Will you use the sentence completion LLaMA model or the image generation model altDiffusion?

You should keep in mind the threats to validity discussed by Blodgett et al. (2021). Make sure your sentences are coherent, grammatical, and target the particular aspect of cultural bias you are interested in.

You will submit your dataset as a single TSV file (tab-separated values). The format will depend somewhat on which prompting paradigm you choose.

You can see examples of my files in the [CS 232 Final Project Resources folder](#). I chose to have 6 conditions for each of my sentences: five countries and a neutral version.

My dataset is formatted as a TSV file with the following four fields:

NUMBER	EXAMPLE	COUNTRY
--------	---------	---------

- NUMBER: An ID number. Each frame sentence should have a unique number.
- EXAMPLE: the sentence itself
- COUNTRY: the name of the country that is being targeted

I first wrote a file containing 32 frame sentences. Then I used a script (`stub_to_prompt.py`) to substitute in the city names for each of the different country conditions. I then manually edited the output to construct the neutral condition and make sure that it sounded natural.

3 Running A Model

I have given you support code for running LLaMA, an open-source large language model, and altDiffusion, an open-source image generation model that supports multiple languages.

I am running both models for you on my research server. You will interact with the model by sending web requests (like in HW 8). **You will need to be on Wellesley wifi or on the Wellesley VPN for this to work.**

3.1 LLaMA

I have written support code for you to query LLaMA in three different modes:

- Completions: receive a completion of your prompt, up to MAX LEN (by default, 10 tokens)
- Words: receive the average probabilities of a list of words, following your prompt
- Tokens: receive the top N most likely next tokens, following your prompt

You can run each of these kinds of queries by calling functions in the `query_LLaMA.py` file. You should import this program at the top of your own program. You can then use each of the three provided functions within your own program:

```
import query_llama

# Retrieve the most likely sequence of next tokens, up to length 5:
print(query_llama.completion_query("My favorite food is",5))

# Retrieve the top 5 most likely tokens and their probabilities:
print(query_llama.token_query("My favorite food is",5))

# Retrieve the average probability of the listed completions:
print(query_llama.word_query("My favorite food is","pickles;pizza;rocks"))
```

The first query shown above, the completion query, will return a completion of the prompt. When I ran it, it was *'chicken and rice.'* Note that this is only 3 words, but it is 5 tokens, because period counts as a token and chicken is split into two subword components ("ch" and "icken").

The second query, the token query, will return a dictionary containing the top 5 most likely next tokens and their probabilities. When I ran it, it produced the following:

```
{'p': 0.15201939642429352, 'ch': 0.0800427719950676,
'a': 0.0690295472741127, 's': 0.04214487597346306,
```

```
'ice': 0.037561580538749695}
```

Note that because of subword tokenization, many of these are not complete words.

The third prompt, the word query, takes a list of words separated by semi-colons, and assesses the likelihood of each to follow the prompt. It returns a dictionary of the words with their probabilities. When I ran it with the word list shown in the code snippet, it returned:

```
{'pickles': 0.20333649714787802, 'pizza': 0.3702385276556015, 'rocks': 0.0}
```

Notice that 'rocks' has zero probability, since it is an unlikely favorite food.

3.2 AltDiffusion

AltDiffusion is an image generation model (like Stable Diffusion, which we used in HW 7). It takes a natural language prompt and produces an image. AltDiffusion works similarly to Stable Diffusion, but is a multilingual model that supports at least 9 languages.

I have written support code for you to query AltDiffusion in **query_altdiffusion.py**. You can import this as a library and call the **generate** function as follows:

```
import query_altdiffusion
```

```
query_altdiffusion.generate("A cat riding a unicycle", "name.jpg")
```

Running this function will create an image file in the directory that your Python file is called from, saved to the filename that is passed in as the second argument to the function.

For instance, the image below is what I generated by running the prompt above:



4 Evaluation Metric

How will you measure the performance of the model with your dataset?

If you are using LLaMA, will you compare the probabilities of sentences? Will you compare the probabilities of specific sentence completions? Will you come up with a metric for evaluating the top-k sentence completions that LLaMA produces?

For my example task, I compared each country-specific sentence to the neutral condition using the top 5 most likely next words according to LLaMA. You can see how I did this in **breakfast_scoring.py** script if you want to do something similar.

You can also come up with other ways of evaluating your task. For instance, you might define a set of words that you think the model might generate and look at the probabilities of these words across versions of your sentences.

If you are generating images, you will need to do a manual annotation step: evaluate your image for some set of properties you're interested in, and come up with a numeric score. Then you should process the scores to compare performance across your conditions. You can see how I did this in **animal_scoring.py**.

Submit a short description of your evaluation metric (1-2 paragraphs), so that I can give you feedback before your final paper is due.

Describe your proposed evaluation metric in detail. You do not need to submit an implemented metric, but you should be clear on how you propose to evaluate the output of the model. What would success look like for your model? How would you identify bias?