Prompt Engineering and Probing with GPT3

For the extra-credit, we will be exploring the recent trend that has revolutionalized this field. With GPT3, we can do a variety of tasks without the need of training a model. All we need to do is convert the task into an text generation task that follows a set of instructions called *prompts*. As an example, the task of sentiment classification can be designed as:

```
Decide whether a Tweet's sentiment is positive, neutral, or negativ
e.

Tweet: I loved the new Batman movie!
Sentiment:
```

The GPT3 model then completes the text above with the response **Positive**. The above prompt is an example of zero-shot prediction, meaning, we are not providing any signal/direction that can guide the decision. We could also design the prompt as follows:

```
Decide whether a Tweet's sentiment is positive, neutral, or negativ
e.

Tweet: I really liked the Spiderman movie!
Sentiment: Positive

Tweet: I loved the new Batman movie!
Sentiment:
```

Now this is an example of 1-shot learning, i.e., you are providing an labeled example of how the output should look and then ask GPT3 to complete the next example. When you use more than 1 labeled example, it is known as few-shot learning. The expectation is that, if you provide more examples in the prompt, it will make better predictions.

Getting Started

In this assignment, we will first need to register for an account at: https://beta.openai.com/ (https://beta.openai.com/) As a free trial, you will get $18 credits to make api calls to the GPT3 server. Once registered, you should go through the docs here: https://beta.openai.com/docs/guides/completion/prompt-design (https://beta.openai.com/docs/guides/completion/prompt-design) to get more info on the capabilities of the model. You can then go directly interact with GPT3 in the playground: https://beta.openai.com/playground (https://beta.openai.com/playground). For making these calls programmatically, we will do the following:

In [12]:
```
# pip install openai
```

In [13]:
```
import os

## Find the API key by clicking on your profile in the openai page. Add the
## Make sure to delete this cell afterwords

os.environ['OPENAI_API_KEY'] = ''
```

In [14]:
```
import os
import openai

openai.api_key = os.getenv('OPENAI_API_KEY')

response = openai.Completion.create(
  model="text-davinci-002",
  prompt="Decide whether a Tweet's sentiment is positive, neutral, or negat
  temperature=0,
  max_tokens=60,
  top_p=1,
  frequency_penalty=0.5,
  presence_penalty=0
)
```

In [15]:
```
response
```

Out[15]:
```
<OpenAIObject text_completion id=cmpl-66PhJU2oLwwnFlf0lA3j9L3dBapTJ at 0x
7f7b0ee38630> JSON: {
  "choices": [
    {
      "finish_reason": "stop",
      "index": 0,
      "logprobs": null,
      "text": " Positive"
    }
  ],
  "created": 1666986769,
  "id": "cmpl-66PhJU2oLwwnFlf0lA3j9L3dBapTJ",
  "model": "text-davinci-002",
  "object": "text_completion",
  "usage": {
    "completion_tokens": 1,
    "prompt_tokens": 31,
    "total_tokens": 32
  }
}
```

In [16]:
```
response['choices'][0]['text']
```

Out[16]:
```
' Positive'
```

If you see ' Positive' as response in the above cell, you have successfully set-up gpt3 in your

system.

Now, the task for the assignment is really just do something cool. For example, you could probe how well GPT3 performs on the tasks in the previous HWs. Or, you could do something like question-answering or summarization, that were not covered in the assignments. The choice is yours.

Submission

Please submit a written report of what task you tried probing, how well did GPT3 do for that task and what were your key takeaways in this experiment.

In HW5, Task A we given two sentences: a premise and a hypothesis, classify the relationship between them. Three relationship: Entailment, contradiction, neutral.

There are three sections in the dataset:

Split sizes (num_samples, num_labels): {'test': (10000, 3), 'train': (550152, 3), 'validation': (10000, 3)}

Example: {'premise': 'A person on a horse jumps over a broken down airplane.', 'hypothesis': 'A person is training his horse for a competition.', 'label': 1}

In hw5, Task B I want to use GPT3 to go further about hw5 task A. In the shared task, our team choose Task 4 human value argument, which is kind like hw5 task A. The shared task given three sentences: a premise, a hypothesis, and a relationship, we need to classify the human value behind these three sentences.

Thre are totally 20 labels (human values):

1. Self-direction: thought
2. Self-direction: action
3. Stimulation
4. Hedonism
5. Achievement
6. Power: dominance
7. Power: resources
8. Face
9. Security: personal
10. Security: societal
11. Tradition
12. Conformity: rules
13. Conformity: interpersonal
14. Humility
15. Benevolence: caring
16. Benevolence: dependability
17. Universalism: concern
18. Universalism: nature
19. Universalism: tolerance

20. Universalism: objectivity

Feature Dataset
ArgumentID Conclusion Stance Premise
A01010 We should prohibit school prayer against it should be allowed if the student wants to pray as long as it is not interfering with his classes
A01011 We should abolish the three-strikes laws in favor of three strike laws can cause young people to be put away for life without a chance to straight out their life
A01012 The use of public defenders should be mandatory in favor of the use of public defenders should be mandatory because some people don't have money for a lawyer and this would help those that don't

Target Dataset
Argument ID Self-direction: thought Self-direction: action Stimulation Hedonism Achievement Power: dominance Power: resources Face Security: personal Security: societal Tradition Conformity: rules Conformity: interpersonal Humility Benevolence: caring Benevolence: dependability Universalism: concern Universalism: nature Universalism: tolerance Universalism: objectivity
A01010 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0
A01011 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1
A01012 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0

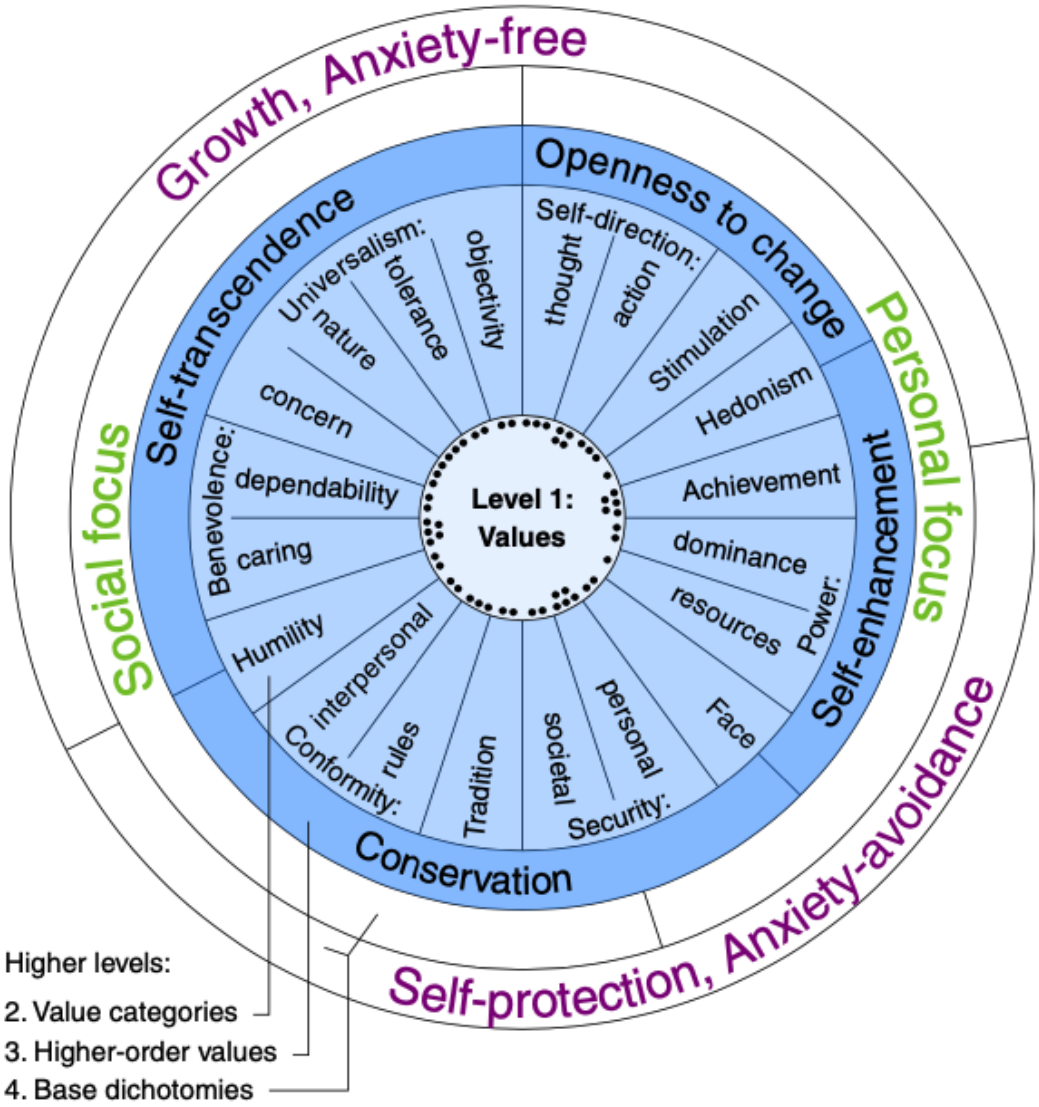
However, the shared task has a big drawback is that it has relatively small dataset, it has only 5220 samples.

So, I decide to enrich the dataset by using GPT3 to manually label human value on the snli dataset from hw5 task A.

I will combine snli dataset (test, train, validation) into 570152 samples, and try to use GPT3 to label one of human values to them.

Here is the example for doing that, I will not process 570152 sample because it will overuse free $18 credits for openai api. So just check if GPT3 can handle this task.

In order to easily interpret the result, I use the higher level label described in the paper to minimize the 20 lables into 4 labels: self-transcendence, openness to change, self-enhancement, conservation.



In [3]:
```
import pandas as pd

labels_df = pd.read_csv('labels-training.tsv', sep='\t')
```

In [4]:
```
# Self-transcendence
labels_df.loc[(labels_df['Universalism: concern'] == 1)
                                        | (labels_df['Universalism: nature'] ==
                                       | (labels_df['Universalism: tolerance'] =
                                       | (labels_df['Universalism: objectivity']
                                       | (labels_df['Benevolence: caring'] == 1)
                                       | (labels_df['Benevolence: dependability'
```

Out[4]:

	Argument ID	Self-direction: thought	Self-direction: action	Stimulation	Hedonism	Achievement	Power: dominance	Power: resources	Fa
6	A01007	0	0	0	0	0	0	0	
7	A01008	0	0	0	0	0	0	0	
8	A01009	0	0	0	0	0	0	0	
9	A01010	1	1	0	0	0	0	0	
10	A01011	0	0	0	0	1	0	0	

5 rows × 21 columns

In [5]:
```
# Conservation
labels_df.loc[(labels_df['Conformity: interpersonal'] == 1)
                                        | (labels_df['Conformity: rules'] == 1)
                                       | (labels_df['Tradition'] == 1)
                                       | (labels_df['Security: personal'] == 1)
                                       | (labels_df['Security: societal'] == 1)]
```

Out[5]:

	Argument ID	Self-direction: thought	Self-direction: action	Stimulation	Hedonism	Achievement	Power: dominance	Power: resources	Fac
0	A01001	0	0	0	0	0	0	0	
1	A01002	0	0	0	0	0	0	0	
3	A01004	0	0	0	0	0	0	0	
4	A01005	0	0	0	0	0	0	0	
5	A01006	0	0	0	0	0	1	0	

5 rows × 21 columns

In [6]:
```
# Self-enhancement
labels_df.loc[(labels_df['Achievement'] == 1)
                                        | (labels_df['Power: dominance'] == 1)
                                       | (labels_df['Power: resources'] == 1)].h
```

Out[6]:

	Argument ID	Self-direction: thought	Self-direction: action	Stimulation	Hedonism	Achievement	Power: dominance	Power: resources	Fa
5	A01006	0	0	0	0	0	1	0	
10	A01011	0	0	0	0	1	0	0	
16	A01017	0	0	0	0	1	0	0	
17	A01018	0	0	0	0	0	1	0	
52	A03013	0	0	0	0	1	1	0	

5 rows × 21 columns

In [8]:
```
# Openness to change
labels_df.loc[(labels_df['Self-direction: thought'] == 1)
                                        | (labels_df['Self-direction: action'] =
                                       | (labels_df['Stimulation'] == 1)].head()
```

Out[8]:

	Argument ID	Self-direction: thought	Self-direction: action	Stimulation	Hedonism	Achievement	Power: dominance	Power: resources	Fa
2	A01003	0	1	0	0	0	0	0	
9	A01010	1	1	0	0	0	0	0	
19	A01020	1	0	0	0	0	0	0	
21	A02002	0	1	0	0	0	0	0	
31	A02012	0	1	0	0	0	0	0	

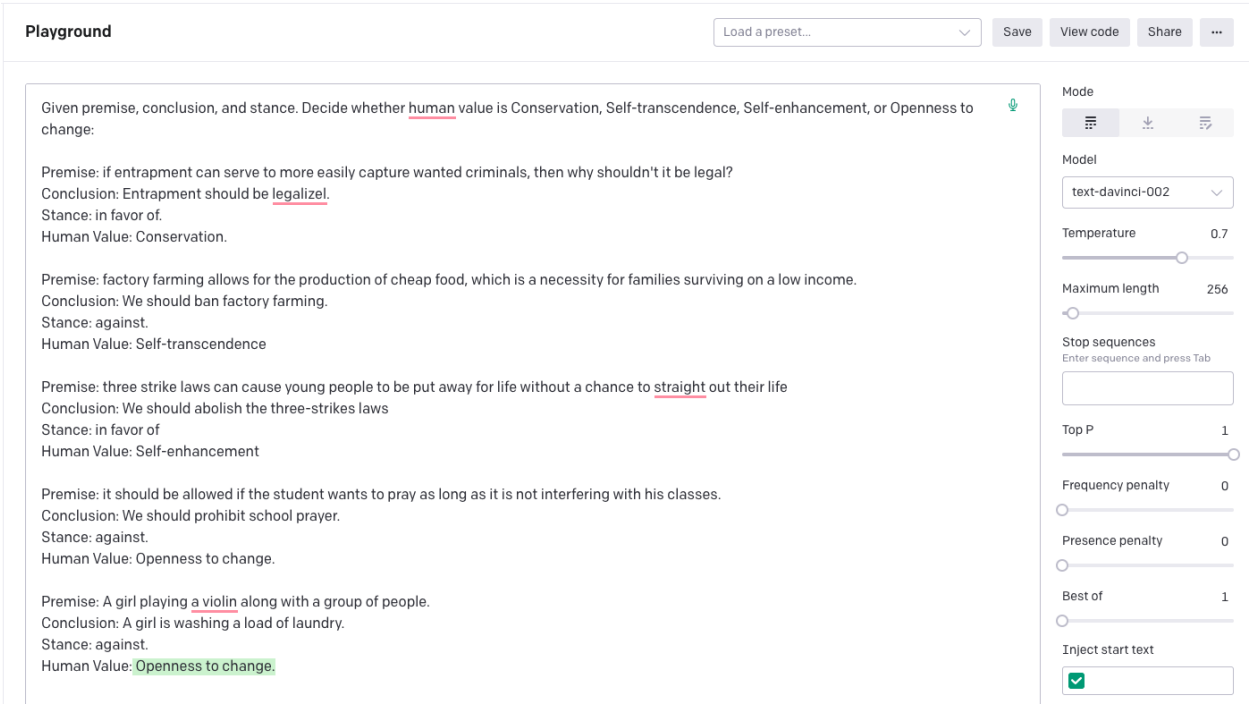
5 rows × 21 columns

zero-shot prediction

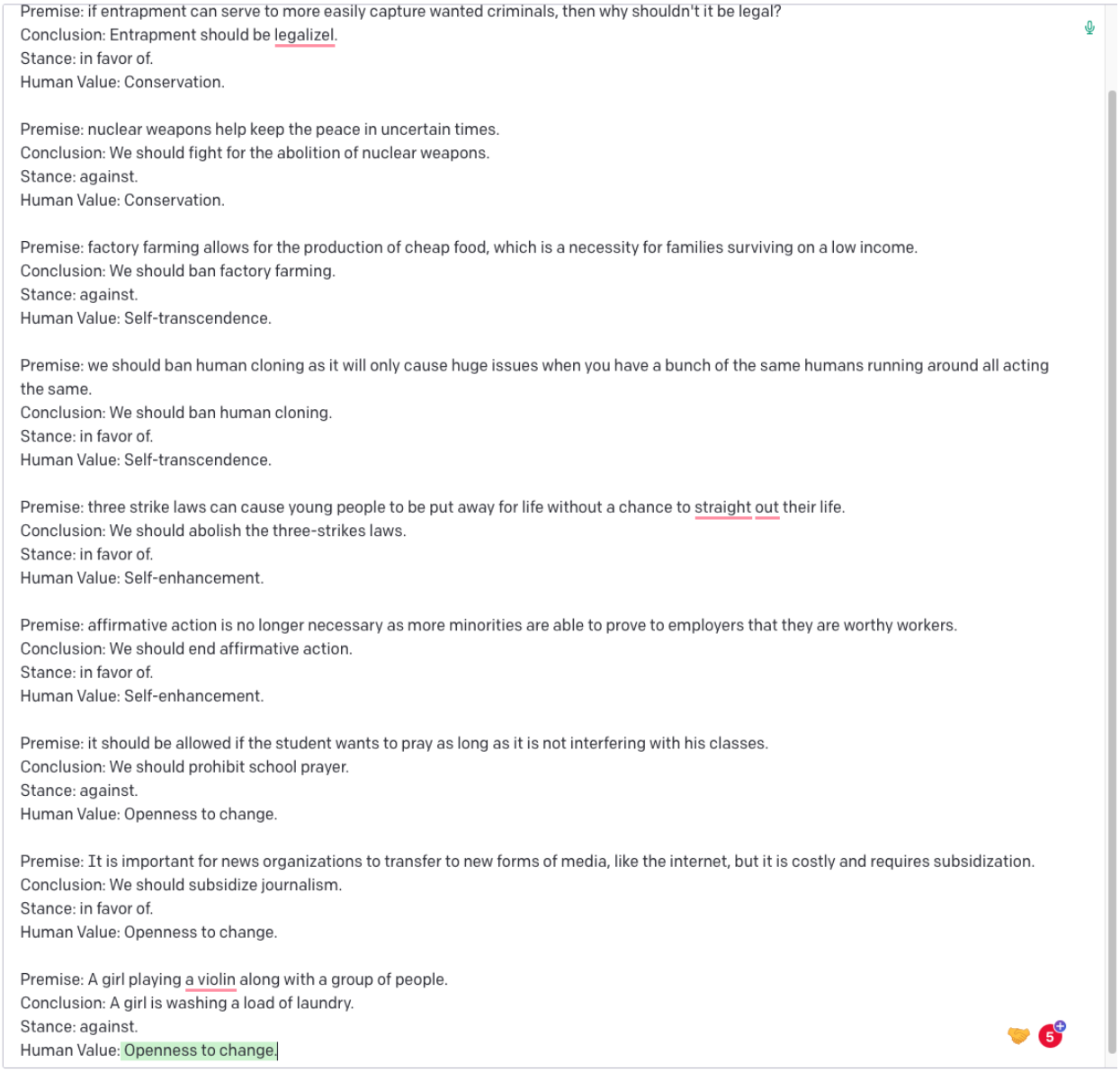
Given premise, conclusion, and stance. Decide whether human value is Conservation, Self-transcendence, Self-enhancement, or Openness to change:

Premise: A girl playing a violin along with a group of people.
Conclusion: A girl is washing a load of laundry.
Stance: against.
Human Value: Self-transcendence

1-shot learning



few-shot learning



It seems that 1 shot and few shot approach with GPT3 gives relatively feasible result but not for zero shot.