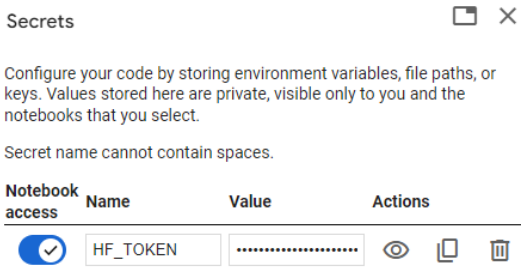
SIOP 2024 Hungry Llama Final Solution Steps

Table of Contents

Task 1: Empathy

Steps

1. Browse to Task 1 Google Colab Python notebook: https://colab.research.google.com/drive/1MaG2JPdBdRJilyyTGSj9to9QuBkV5Kb4?usp=sharing
2. Setup Hugging Face API token in Google Colab (requires login):



   a. **Name:** HF_TOKEN
   b. **Value:** TBD
   c. Allow notebook access to token
3. Manually upload just the empathy test data file (requires login):
   a. empathy_test_public.csv
4. Run Colab notebook
   a. Runtime | Run All
      i. *Note, this may take 45-60 minutes to complete*
5. Download empathy test submission output file for merging later:
   a. empathy_submit_test.csv

LLMs Used:

Mistral-7B-Instruct-v0.2

Mixtral-8x7B-Instruct-v0.1

LLM Parameters:

```
prompt_template = {
    "inputs": f"<s>[INST] {prompt}[/INST]",
    "parameters": {
        "do_sample": False,
        "max_new_tokens": 2000,
        "top_p": 0.9,
        "top_k": 1,
        "repetition_penalty ": 1,
        "presence_penalty": 0,
        "frequency_penalty": 0,
        "return_full_text": False
    }
}
```

LLM Prompts:

```
empathy_prompt = f"""You are an email message categorization bot. Your task is to categorize the email message after <<<>>> into one of the following predefined categories:

A
B
C

You will only respond with a JSON object with the category A or B, the probability of being category A, and confidence. Do not provide explanations or notes. If you're not sure use category C.

###
Here are some examples, but ignore the order:

{empathetic_examples}{unempathetic_examples}###
"""

empathy_prompt += """<<<
Email Message: {0}
>>> """
```

```
sentiment_prompt = """Rate, from 0 to 1, the overall sentiment towards the recipient in the following email message. You will only respond with a JSON object with the sentiment rating, and confidence. Do not provide explanations or notes.

{0}
"""

negativity_prompt = """Rate, from 0 to 1, the sender's overall negativity towards the recipient in the following email message. You will only respond with a JSON object with the negativity rating, and confidence. Do not provide explanations or notes.

{0}
"""

approval_prompt = """Rate, from 0 to 1, the sender's overall approval for the recipient in the following email message. You will only respond with a JSON object with the approval rating, and confidence. Do not provide explanations or notes.

{0}
"""

appreciation_prompt = """Rate, from 0 to 1, the sender's appreciation towards the recipient's contributions to the project in the following email message. You will only respond with a JSON object with the appreciation rating, and confidence. Do not provi

{0}
"""

support_prompt = """Rate, from 0 to 1, how well the sender offers support in terms of learning resources, mentorship or coaching for the recipient in the following email message. You will only respond with a JSON object with the support rating, and confi

{0}
"""

rapport_prompt = """Rate, from 0 to 1, how well the sender builds rapport through shared experiences with the recipient in the following email message. You will only respond with a JSON object with the rapport rating, and confidence. Do not provide expla

{0}
"""

phrase_prompt = """Rate, from 0 to 1, the number of empathetic phrases in the following email message. You will only respond with a JSON object with the phrase rating, and confidence. Do not provide explanations or notes.

{0}
"""
```

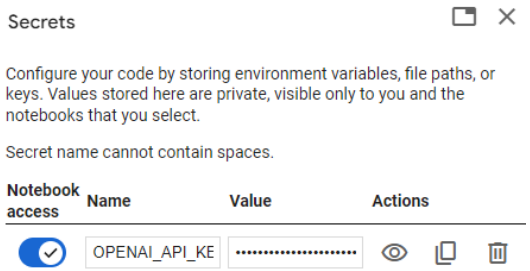
LLM Usage:

Multiple LLM prompts were used to evaluate empathy for the given text. The results were weighted and summed to produce a final empathy classification.

Task 2: Interview

Steps

1. Browse to Task 2 Google Colab Python notebook: https://colab.research.google.com/drive/1G8lJJcrB273jK-kfpZg8Z5n6aPJuXbpr?usp=sharing
2. Setup OpenAI API token in Google Colab (requires login):



   a. **Name:** OPENAI_API_KEY
   b. **Value:** TBD
   c. Allow script access to token
3. Manually upload interview and personality, training and test data files (requires login):
   a. interview_train.csv
   b. interview_test_public.csv
   c. personality_train.csv
   d. personality_test.csv
4. Run Colab notebook
   a. Runtime | Run All
5. Download interview test submission output file for merging later:
   a. task2output.csv

LLM Used:

ChatGPT 4

LLM Parameters:

```
def openai_api_query(user_prompt, system_prompt):
  response = client.chat.completions.create(
    model="gpt-4",
    messages=[
      {"role": "system", "content": f"{system_prompt}"},
      {"role": "user", "content": f"{user_prompt}"},
    ],
    temperature=0,
    max_tokens=2000,
    top_p=1,
    frequency_penalty=0,
    presence_penalty=0,
    seed=42
  )
```

LLM Prompt:

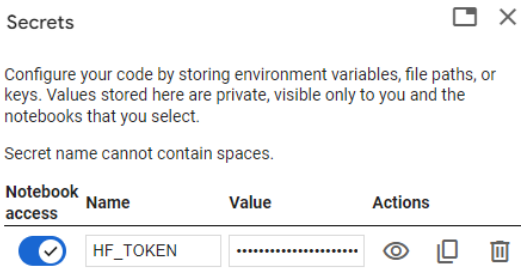
```
  system_prompt = f"""Given the following question and response examples below, learn the writing style of the responses:
#####
Question 1:
{q1}
Response 1:
{a1}

Question 2:
{q2}
Response 2:
{a2}

Question 3:
{q3}
Response 3:
{a3}
#####

Generate a response for the provided question following these rules:
- Start with a restatement of the question
- Have a similar writing style
- Write at least {avg_sentences} sentences
- Be similar in tone{extraversion_rule}{agreeableness_rule}{conscientiousness_rule}{emotional_stability_rule}{openness_rule}{flesch_rule}
"""
```

LLM Usage:

Previous question responses and OCEAN personality traits were used as input into the prompt template for ChatGPT-4 to perform text generation.

Task 3: Clarity

1. Browse to Task 3 Google Colab Python w/R notebook: https://colab.research.google.com/drive/1T356RLQSG4yAqx1Qcn5G9V8RDEgV2rni?usp=sharing
2. Setup Hugging Face API token in Google Colab (requires login):



   a. **Name:** HF_TOKEN
   b. **Value:** TBD
   c. Allow script access to token
3. Manually upload clarity training and test data files (requires login):
   a. clarity_train.csv
   b. clarity_test_public.csv
4. Run Colab notebook
   a. Runtime | Run All
      i. *Note, this may take 45-60 minutes to complete*
5. Download clarity test submission output file for merging later:
   a. clarity_test1.csv

LLMs Used:

Mixtral-8x7B-Instruct-v0.1

bart-large-mnli

LLM Parameters:

```
call_to_mixtral <- function(prompt) {

  "https://api-inference.huggingface.co/models/mistralai/Mixtral-8x7B-Instruct-v0.1" |>
    request() |>
    req_auth_bearer_token(token) |>
    req_body_json(
      list(
        inputs = prompt,
        parameters =
          list(
            do_sample = FALSE,
            return_full_text = FALSE
          )
      )
    ) |>
    req_perform() |>
    resp_body_json(simplifyVector = TRUE)
```

```
call_on_bart <- function(prompt) {

  "https://api-inference.huggingface.co/models/facebook/bart-large-mnli" |>
    request() |>
    req_auth_bearer_token(token) |>
    req_body_json(
      list(
        inputs = prompt,
        parameters =
          list(
            candidate_labels =
              c("openness", "conscientiousness", "extraversion",
                "agreeableness", "neuroticism")
          )
      )
    ) |>
    req_perform() |>
    resp_body_json(simplifyVector = TRUE)

}
```

LLM Prompts:

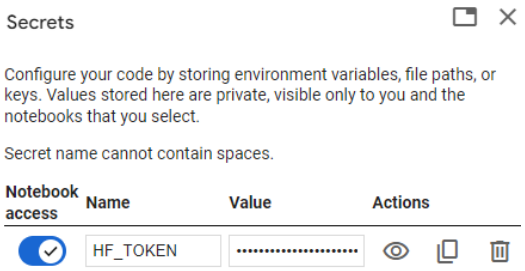
```
  str_c(
    "<s> [INST] -- context -- Learn from the examples. ",
    "Respond with a number between 1 and 7. Do not explain your reasoning.",
    " -- examples -- ",
    str_c(pre_prompt, collapse = " "),
    "</s>",
    " [INST] ", focal[[1]], " [/INST]"
  )

}
```

LLM Usage:

Mixtral was used to rate all clarity examples in the training and test data. The LLM results were combined with NLP features as input into a machine learning model used to predict the final clarity rating.

Task 4: Fairness

1. Browse to Task 4 Google Colab Python w/R notebook: https://colab.research.google.com/drive/1ONDm_PalPnt91ad_TwDS-ROH7F2wLVAs?usp=sharing
2. Setup Hugging Face API token in Google Colab (requires login):



   a. **Name:** HF_TOKEN
   b. **Value:** TBD
   c. Allow script access to token
3. Manually upload fairness training and test data files (requires login):
   a. fairness_train.csv
   b. fairness_test_public.csv
4. Run Colab notebook
   a. Runtime | Run All
5. Download fairness test submission output file for merging later:
   a. fairness_test1.csv

LLM Used:

Mixtral-8x7B-Instruct-v0.1

LLM Parameters:

```
call_to_mixtral <- function(prompt) {

  "https://api-inference.huggingface.co/models/mistralai/Mixtral-8x7B-Instruct-v0.1" |>
    request() |>
    req_auth_bearer_token(token) |>
    req_body_json(
      list(
        inputs = prompt,
        parameters =
          list(
            do_sample = FALSE,
            return_full_text = FALSE
          )
      )
    ) |>
    req_perform() |>
    resp_body_json(simplifyVector = TRUE)
```

LLM Prompts:

```
          return(
            str_c(
              "Question ", d, ": Which option is fairer?\n\n",
              "Option 1 = ", a, "\nOption 2 = ", b,
              "\n\nAnswer ", d, ": ",
              switch(c, "first" = "Option 1", "second" = "Option 2")
            )
          )

        }
      )

    str_c(
      " ---- Context ---- \n\n",
      str_c(pre_prompt, collapse = "\n\n"),
      "\n\n---- Question for Mistral ---- \n\n",
      "Answer Question 24 below given the responses to Questions 1 ",
      "through 23 above. Only respond with 'Option 1' or 'Option 2' ",
      "and do not explain your reasoning. ",
      "The order of 'Option 1' and 'Option 2' is random. Do not use the order ",
      "that the options appear when judging which is fairer. ",
      "The order of Questions 1 to 23 is random. ",
      "Do not use the order in which the questions to judge the fairness of ",
      "the options.",
      "\n\n",
      "Question 24: Which option is fairer?\n\n",
      "Option 1 = ", focal[[1]],
      "\nOption 2 = ", focal[[2]], "\n\n",
      "Answer 24:"
    )
```

LLM Usage:

The LLM was used to determine which option was fairer given examples from the training set

Last Step: Merge Individual Task Results

1. Browse to “Merge” Google Colab Python notebook: https://colab.research.google.com/drive/1TmZwPjbPiyIuwhTA2fTfPBhuTU2hqH3M?usp=sharing
2. Manually upload the four separate task output files:
    a. empathy_submit_test.csv
    b. task2output.csv
    c. clarity_test1.csv
    d. fairness_test1.csv
3. Run Colab notebook
    a. Runtime | Run All
4. Download merged submission output file:
    a. test_submit_merged.csv