Machine Learning Competition

SIOP ANNUAL CONFERENCE

CHICAGO and ONLINE • April 17-20, 2024

Mustafa Akben, PhD

Assistant Professor of Management

Aaron Satko

Computer Science & AI Club Co-Chair

ELON UNIVERSITY

Four tasks, Four Challenges

- **Empathy Prediction** — Binary Classification
- **Item Clarity** — Correlational
- **Interview Completion** — Cosine Similarity
- **Policy Fairness** — Binary Classification

Task 1: Empathy Prediction

- Zero-Shot COT
- Syntactic chain-of-though with N-shot
- Context Learning + Text Completion + Self-Consistency
- Label Learning Model
- Elo Rating for Empathy Ratings

- ***Gemini 1.5* | *GPT 3.5 - 4* | *Claude3x***

Context Learning + Text Completion + Self-Consistency

- Learn from the context inputs
- Complete text without any prompt with a foundational model
  - Bison from Palm | Davinci-002 from GPTs | Gpt3.5-instruct from GPTs
- Repeat them N times (N is an odd number to get mode)
- Get the mode of the predictions

- **Accuracy at train : .60 | Dev = .57**

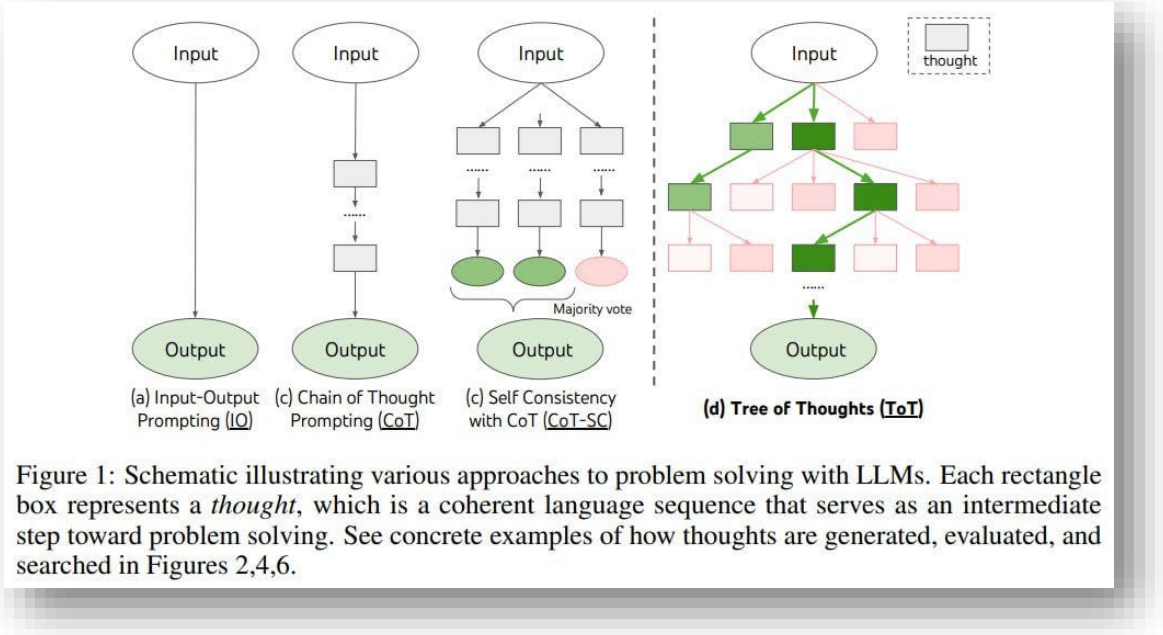
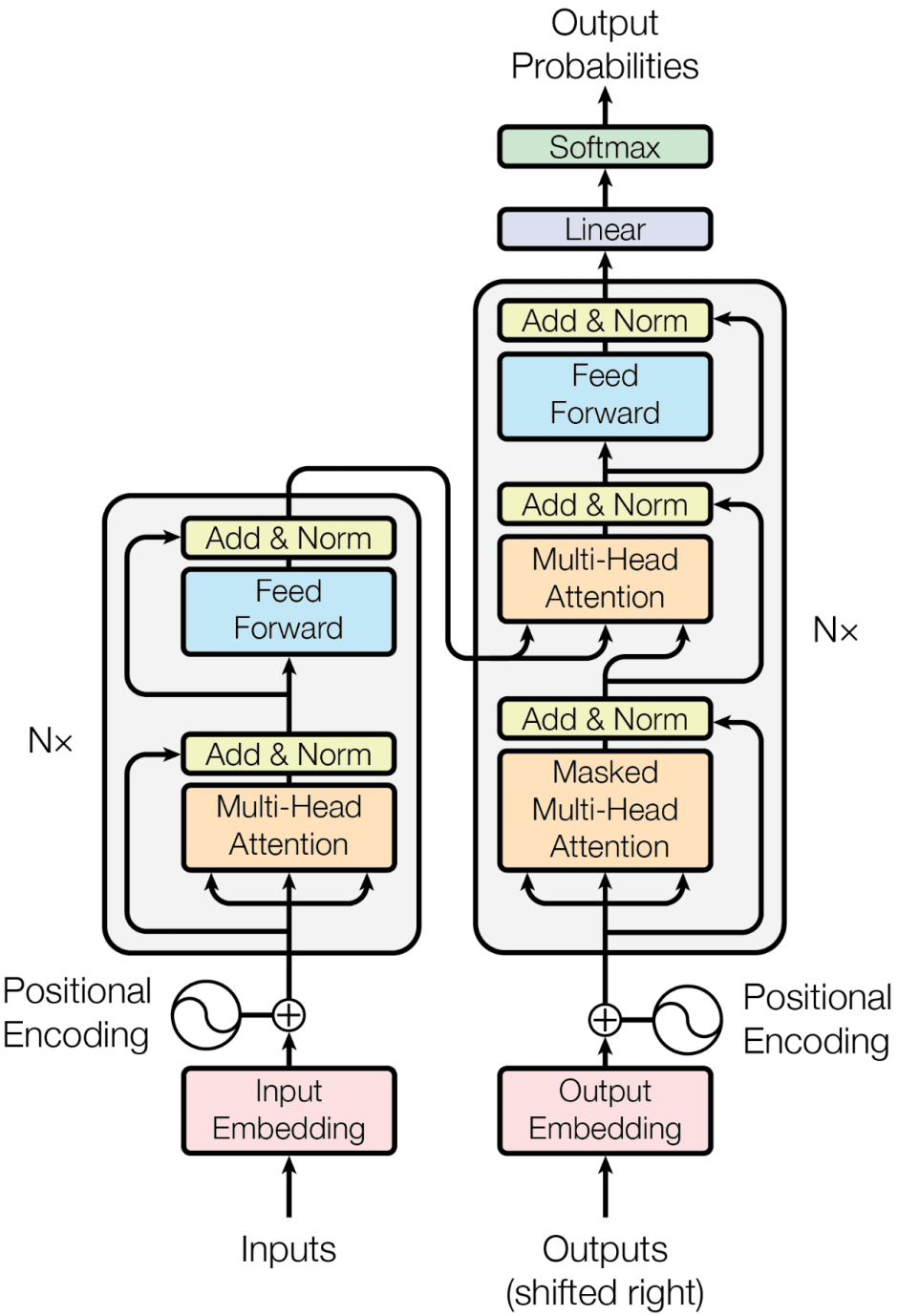
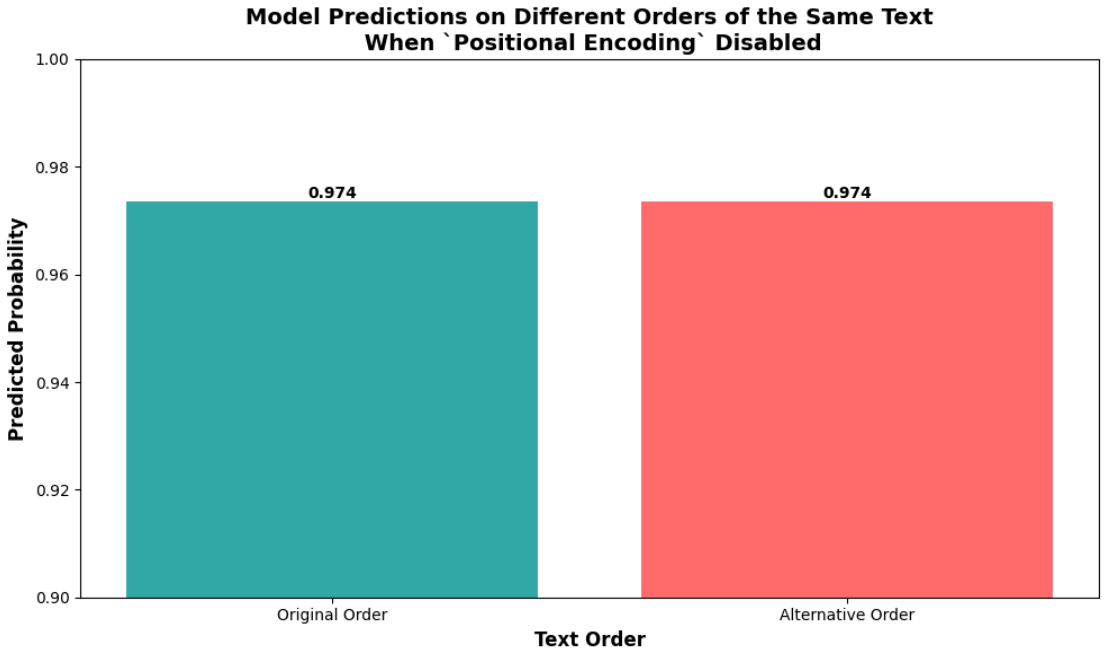


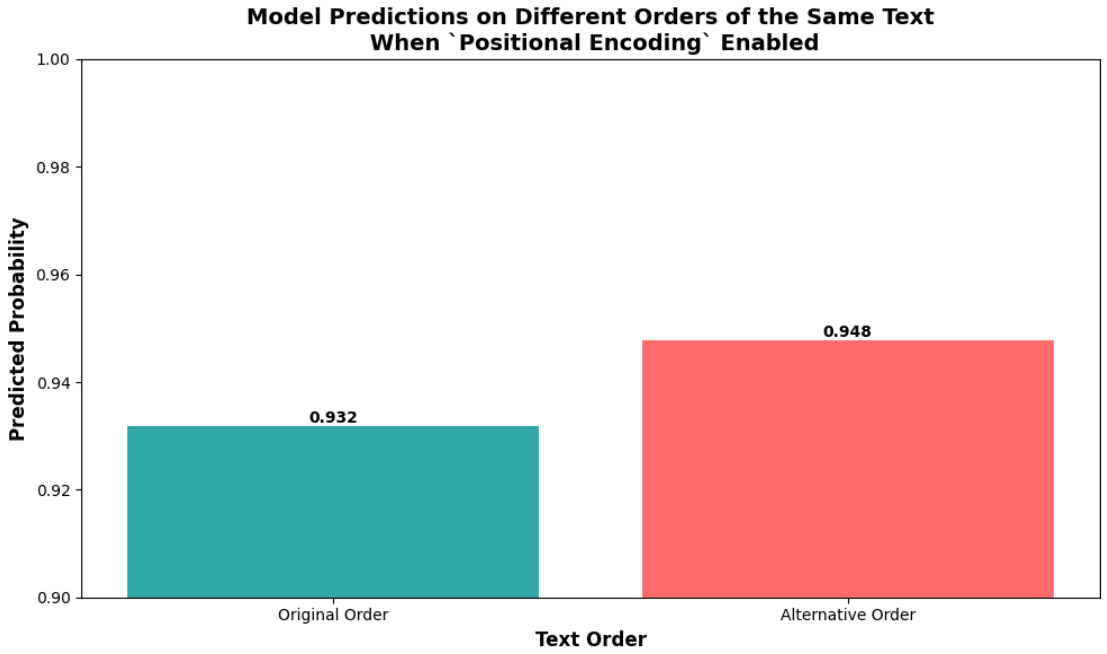
Figure 1: Schematic illustrating various approaches to problem solving with LLMs. Each rectangle box represents a *thought*, which is a coherent language sequence that serves as an intermediate step toward problem solving. See concrete examples of how thoughts are generated, evaluated, and searched in Figures 2,4,6.


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

But why?

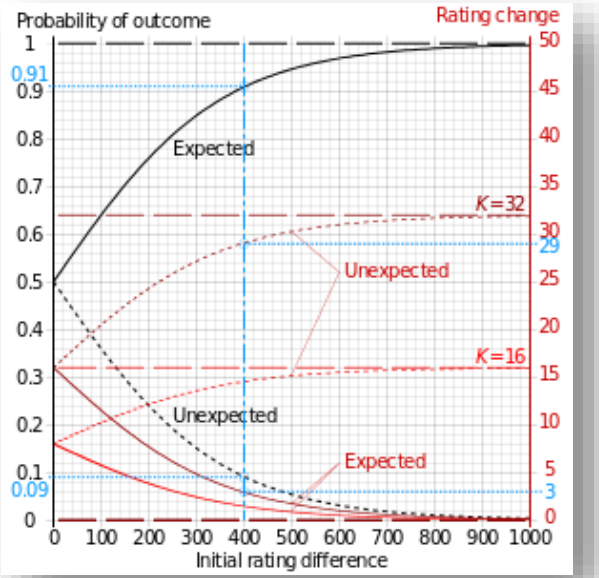
```python
class TransformerBlock(layers.Layer):
    def __init__(self, embed_dim, num_heads, ff_dim, rate=0.1):
        super().__init__()
        self.att = layers.MultiHeadAttention(num_heads=num_heads, key_dim=embed_dim)
        self.ffn = keras.Sequential(
            [layers.Dense(ff_dim, activation="relu"), layers.Dense(embed_dim),]
        )
        self.layernorm1 = layers.LayerNormalization(epsilon=1e-6)
        self.layernorm2 = layers.LayerNormalization(epsilon=1e-6)
        self.dropout1 = layers.Dropout(rate)
        self.dropout2 = layers.Dropout(rate)

    def call(self, inputs):
        attn_output = self.att(inputs, inputs)
        attn_output = self.dropout1(attn_output)
        out1 = self.layernorm1(inputs + attn_output)
        ffn_output = self.ffn(out1)
        ffn_output = self.dropout2(ffn_output)
        return self.layernorm2(out1 + ffn_output)


class TokenAndPositionEmbedding(layers.Layer):
    def __init__(self, maxlen, vocab_size, embed_dim,position_embed=True):
        super().__init__()
        self.token_emb = layers.Embedding(input_dim=vocab_size, output_dim=embed_dim)
        self.pos_emb = layers.Embedding(input_dim=maxlen, output_dim=embed_dim)
        self.position_embeds = position_embed

    def call(self, x):
        maxlen = ops.shape(x)[-1]
        positions = ops.arange(start=0, stop=maxlen, step=1)
        positions = self.pos_emb(positions)
        x = self.token_emb(x)
        if self.position_embeds:
          return x + positions
        else:
          return x
```


Model Predictions on Different Orders of the Same Text
When `Positional Encoding` Disabled
Predicted Probability
1.00
0.98
0.96
0.94
0.92
0.90
0.974
0.974
Original Order
Alternative Order
Text Order

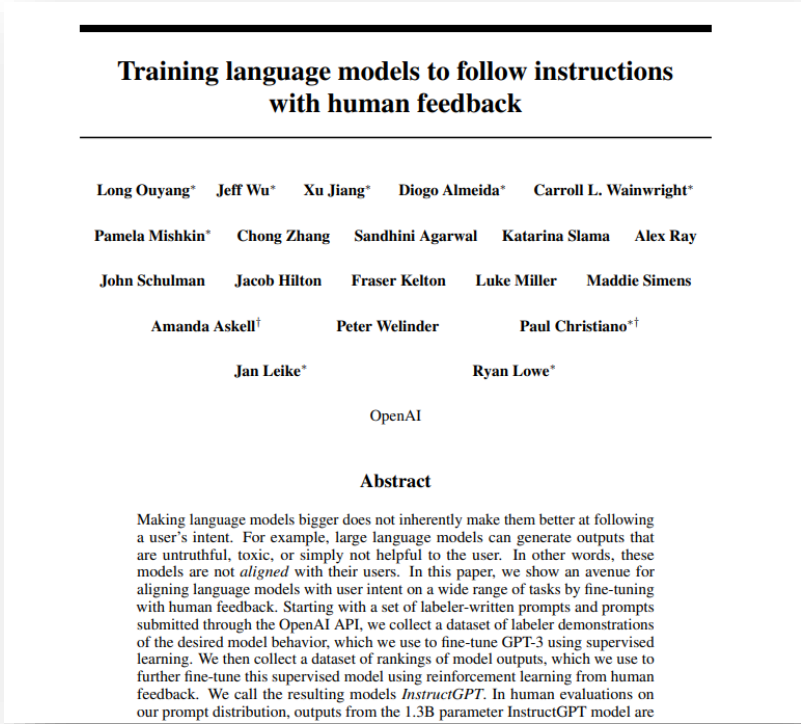

Model Predictions on Different Orders of the Same Text
When `Positional Encoding` Enabled
Predicted Probability
1.00
0.98
0.96
0.94
0.92
0.90
0.932
0.948
Original Order
Alternative Order
Text Order

Elo Ratings for Classification

- Elo ratings is initially used in chess competitions
- LLM & RLHF
- Unexpected results change the ratings



$$E_a = \frac{1}{1 + 10^{\frac{R_b - R_a}{400}}}$$



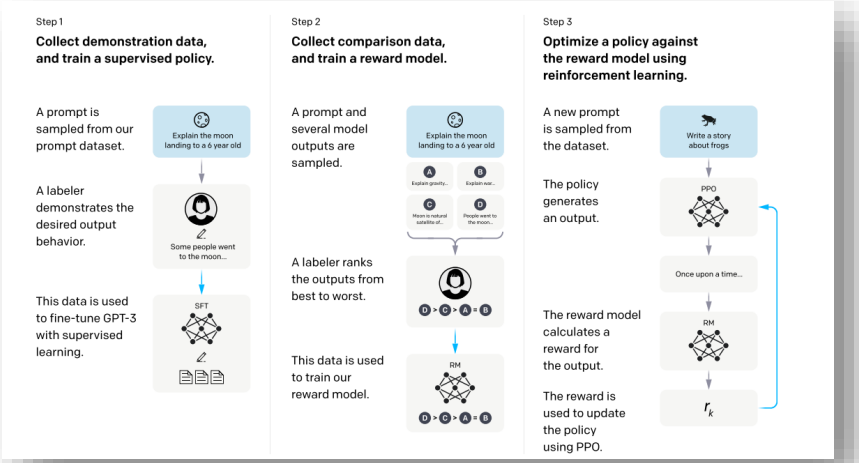
Training language models to follow instructions with human feedback

Long Ouyang* Jeff Wu* Xu Jiang* Diogo Almeida* Carroll L. Wainwright*
Pamela Mishkin* Chong Zhang Sandhini Agarwal Katarina Slama Alex Ray
John Schulman Jacob Hilton Fraser Kelton Luke Miller Maddie Simens
Amanda Askell† Peter Welinder Paul Christiano*†
Jan Leike* Ryan Lowe*

OpenAI

Abstract

Making language models bigger does not inherently make them better at following a user's intent. For example, large language models can generate outputs that are untruthful, toxic, or simply not helpful to the user. In other words, these models are not *aligned* with their users. In this paper, we show an avenue for aligning language models with user intent on a wide range of tasks by fine-tuning with human feedback. Starting with a set of labeler-written prompts and prompts submitted through the OpenAI API, we collect a dataset of labeler demonstrations of the desired model behavior, which we use to fine-tune GPT-3 using supervised learning. We then collect a dataset of rankings of model outputs, which we use to further fine-tune this supervised model using reinforcement learning from human feedback. We call the resulting models *InstructGPT*. In human evaluations on our prompt distribution, outputs from the 1.3B parameter InstructGPT model are



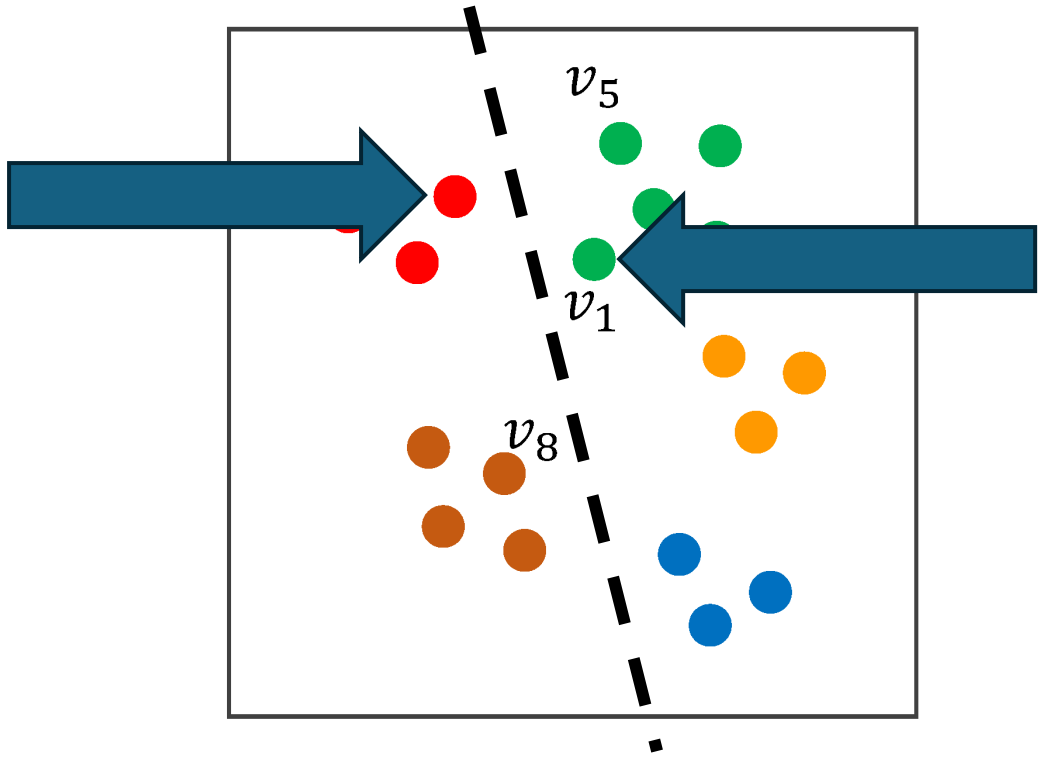
Elo Rating | Class X

“Hi Jonathan, I hope this message finds you well. I hear things are going well with the Beta project. That said, Terry mentioned that there were some issues with the reports. From what I understand, they would like them to be more concise and straight to the point, as well as more business focused. I recommend you reach out to Terry so you both could review in detail one of the reports he submits. This should help you help you align to their expectations. Additionally, I'd be happy to review the reports before you send them off to Terry and provide my feedback. I know this project is important to you, so please let me know how this meeting goes and how else I can help. Regards, William”

Elo Rating | Class X

“Hi Jonathan, First off i'd like to say i'm very pleased to see your enthusiasm towards your work with ABC-5, I appreciate your commitment to doing your best work for the company. I would like to offer some insight into your work that should assist you in achieving your goals. We all need to be continually improving to move forward and I believe I can provide you with some assistance in this area.In regards to your report writing, i'd like to to understand that although your reports cover a lot of good information, we need to keep in mind the target audience and the key objectives of the information. The people that read these need to be able to digest the critical points and understand the facts, so try to find the best way to make the information concise and strictly factual. In addition to this, reports are much more useful when the structure or flow of the report is orderly, it asissts the reader in understanding the information within and allows them to make informed decisions based on your data. It may be a good idea to ask your manager for training opportunities for report writing to develop your skills even further, and be sure to reguarly check in with your manager to make sure your work is aligned with the objectives of the team.I hope these points help you in your job. Please keep up the great work! Thank you!”

Elo Ratings Classification



Elo Rating

1- Sample **two** semantically similar content

2- Ask GPT4 model to choose one

3- For the chosen one, Win rate +1

4 – Calculate the Elo Ratings

5 – Repeat this comparative game N times

6- Transform the Elo Ratings to probs

Prompt

Which feedback would you prefer? You will pick the feedback that makes you feel motivated to work harder and understood, rather than being judge. Return either `Text 1` OR `Text 2`

Text 1

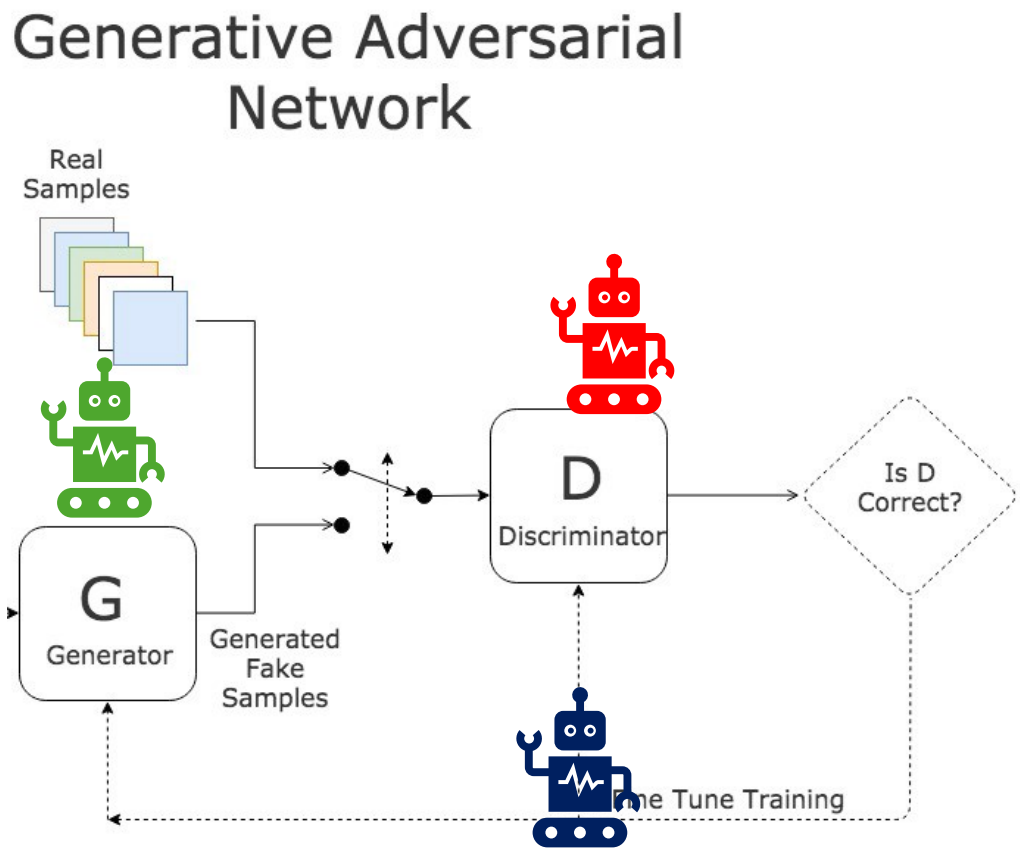
#Text 2

Accuracy in train based on the games: .70 | Dev = .63-.65

Label Learning Model

```
You will be classifying a job candidate's feedback into one of two classes based on similarity to provided example feedbacks for each
class.

The two classes are:
<class1>{$CLASS1}</class1>
<class2>{$CLASS2}</class2>

Here are example feedbacks for <class1>:
<example1>{$EXAMPLE1}</example1>
<example2>{$EXAMPLE2}</example2>
<example3>{$EXAMPLE3}</example3>

Here are example feedbacks for <class2>:
<example4>{$EXAMPLE4}</example4>
<example5>{$EXAMPLE5}</example5>
<example6>{$EXAMPLE6}</example6>

Now, here is the feedback to classify:
<feedback>
{$FEEDBACK}
</feedback>

Carefully read the above feedback. Compare it to the provided examples for each class.

<reasoning>
Write out your reasoning for which class you think the feedback belongs to here. Explain which
examples it is most similar to and why. Note any specific details or phrases that influenced your
decision.
</reasoning>

<classification>
Now output the class you believe the feedback falls into, either <class1> or <class2>.
</classification>
```

Experimenting with GAN – LLM Agent Models



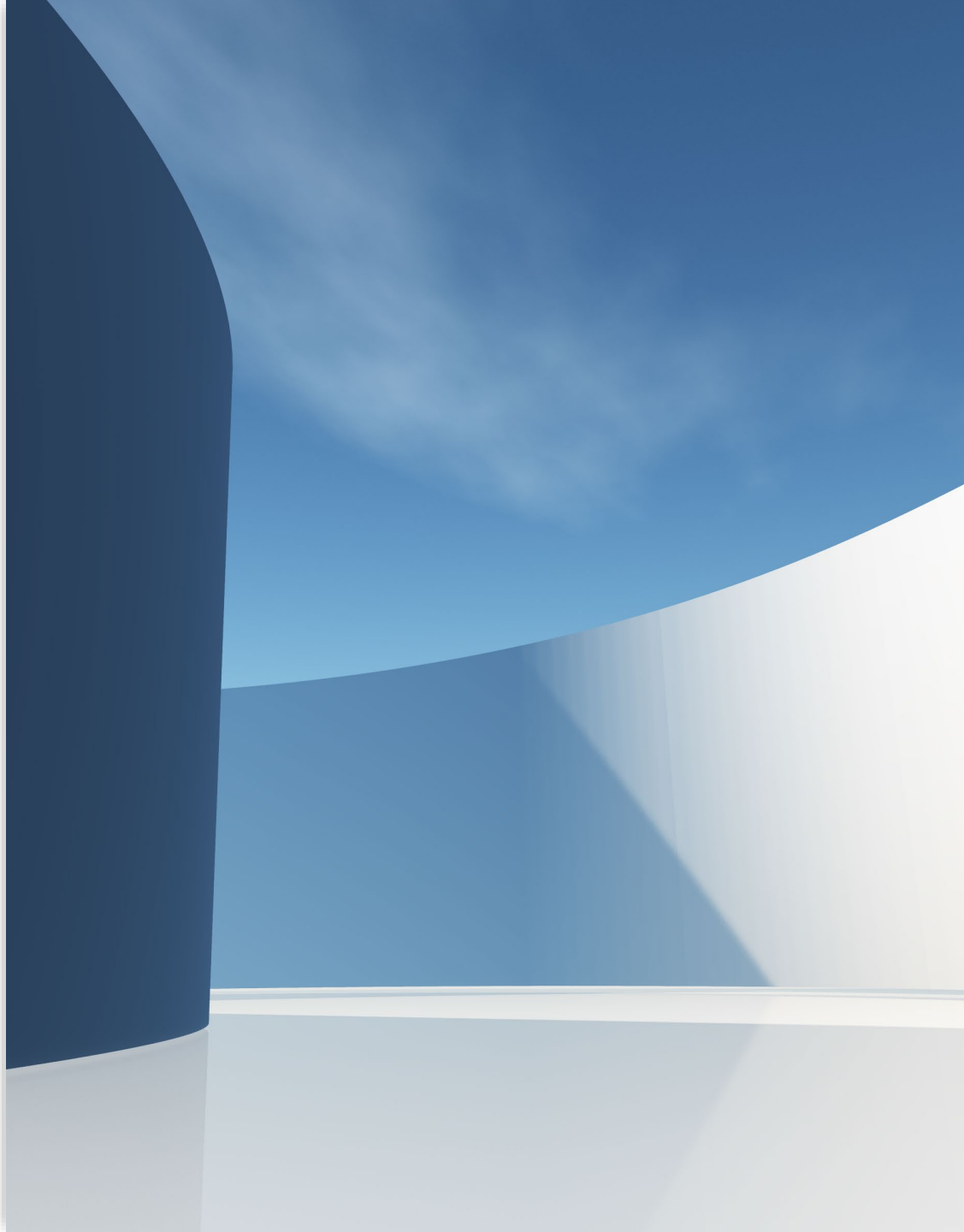
Final Empathy Prediction

- Ensembled all prediction from
  - Text completion N-shot
  - Elo Rating
  - Label Learning
- Majority Votes

- **Train Accuracy = .66-.70**
- **Dev = .60 - .63**

Task 2: Item Clarity

- Label Learning with Meta-Prompt (Latent Space Learning)
- Generate N – Expert for each sub dimension (Binary) for each model
  - GPT4
  - GPT3.5
  - Claude3
- Aggregate the average ratings as an expert ratings from the models
- Use a final ensembled rating for the full text with a stronger model



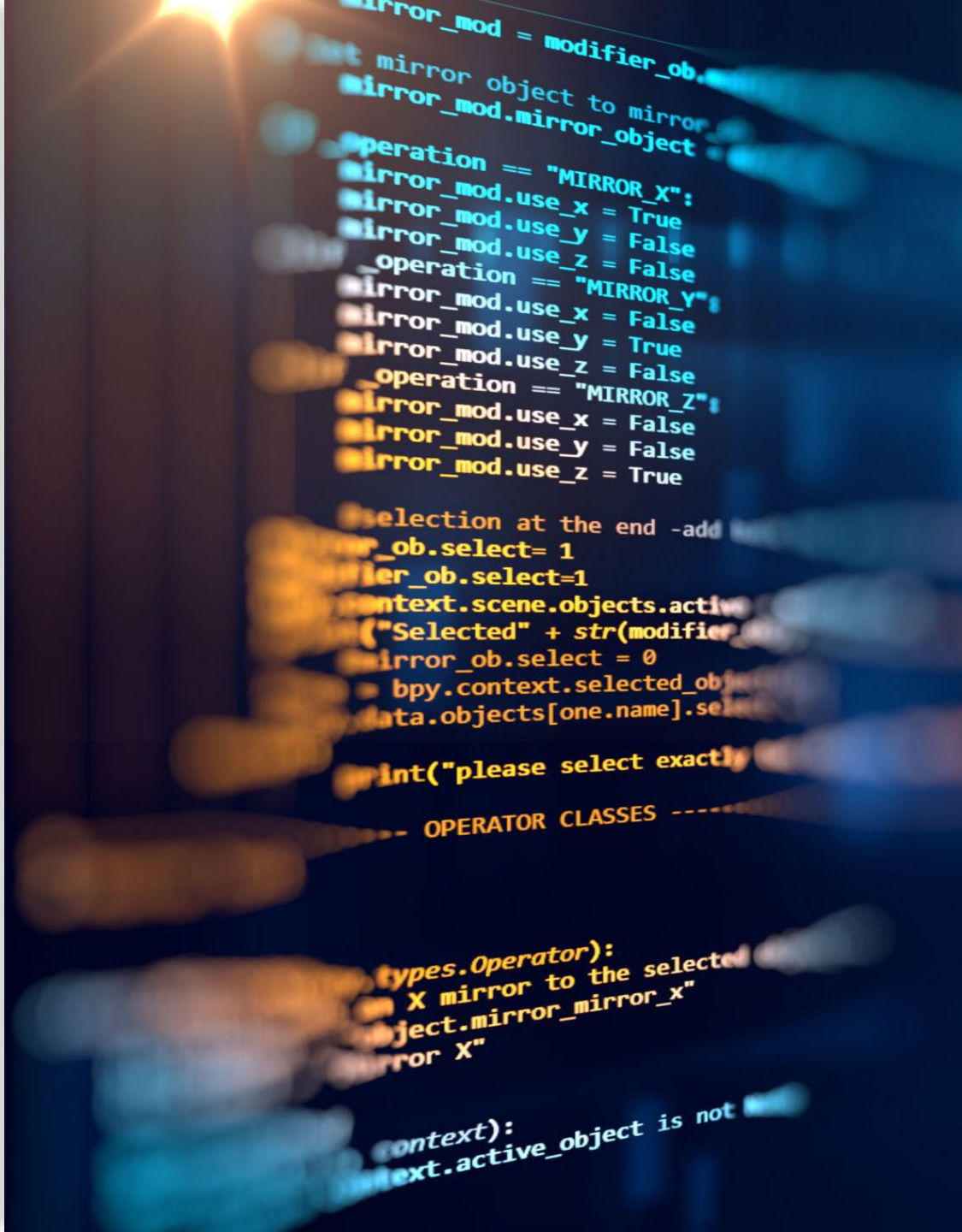
Meta-Prompt

You will be analyzing a set of survey item texts and their corresponding clarity scores to identify patterns that distinguish items that received high clarity scores from those that received low scores.

Here are the survey item texts:

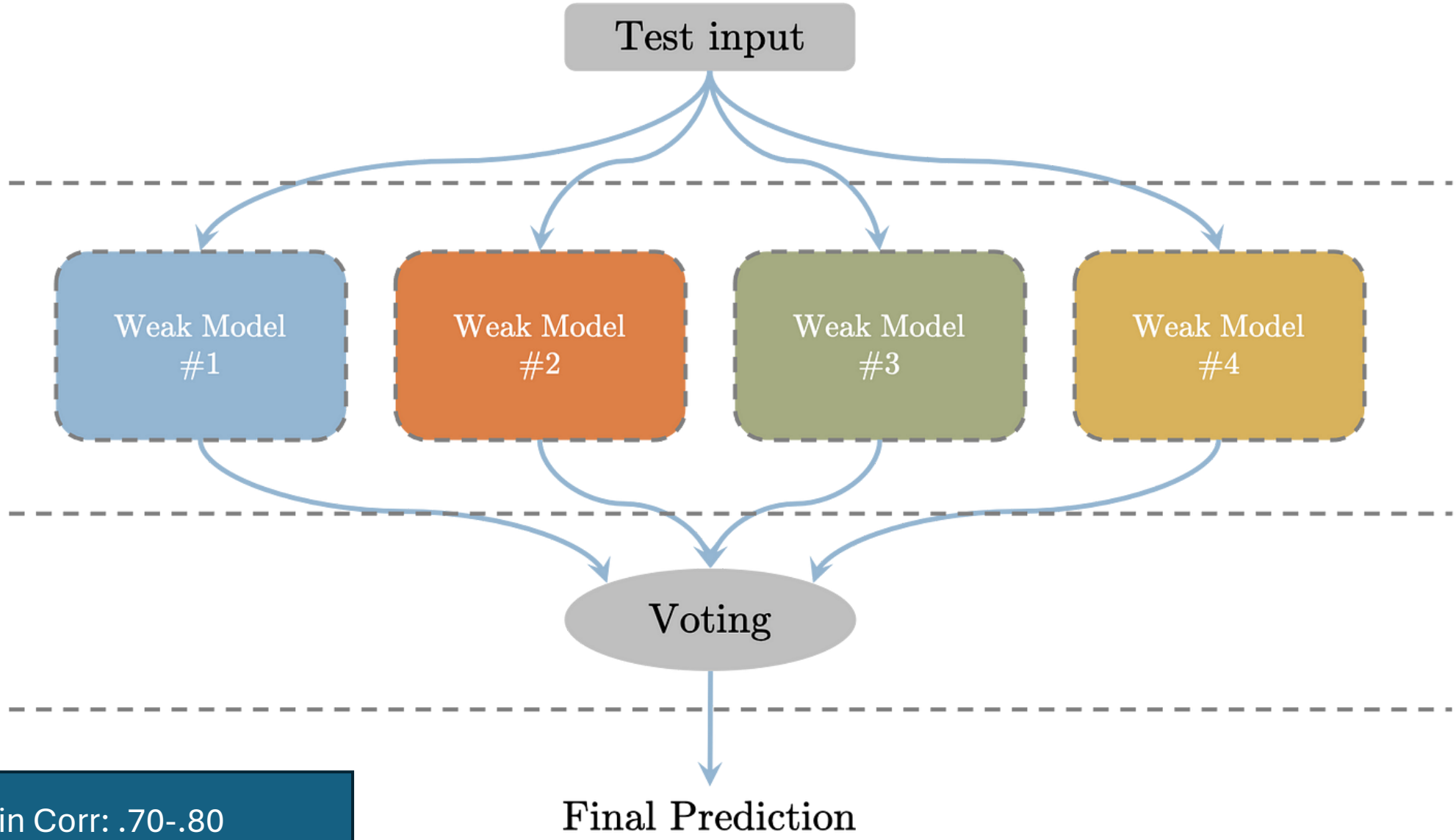
{text}

Please generate binary rules for the classes.



Binary Questions

```
prompt_mapping = {
        'passive': "Is this text written in a passive voice, such as the cake was baked by Emily, or I am considered by my peers to be talented, or etc? Re
        'negative': "Is this text negatively worded such as `not`, `none` etc.? Return only Yes or No.\n",
        'compound': "Does this text use a compound sentence? Return only Yes or No.\n",
        'complex': "Does this text use a complex sentence? Return only Yes or No.\n",
        'simple': "Does the text use simple, everyday language that a lay person can understand? Return only Yes or No.\n",
        'jargon': "Is the text free of jargon or technical terms? Return only Yes or No.\n",
        'focused': "Does the text express a single, focused thought or action? Return only Yes or No.\n",
        'vague': "Is the text free of vague or ambiguous wording? Return only Yes or No.\n",
        'active': "Does the text use strong, active verbs? Return only Yes or No.\n",
        'easy': "Is the text easy to read and understand on the first pass? Return only Yes or No.\n",
        'easy2': "Is the text easy to read and understand for a third grader? Return only Yes or No.\n",
        'unnecessary': "Is the text free of unnecessary words or phrases? Return only Yes or No.\n",
        'double': "Does the text contains `double negatives` or `convoluted phrasing` such as `I do not dislike` or so on? Return only Yes or No.\n",
        'vocabulary': "Is the vocabulary of the text appropriate for a high school educated reader? Return only Yes or No.\n",
        'clarity': "Is this text clear for a third-grader to read? Return only Yes or No.\n"
    }
```



Train Corr: .70-.80
Dev Corr: .60-.70

Task 3: Interview Completion

- Gpt4 Model
- Instructed to complete task based on the style, tone, and important characteristics
- Generate N ***in-context*** examples
  - In-context so that model generate more diverse examples
- Selected the completion that has the highest cosine similarity with the input

- **Cosine Similarity: .50 - .53**

Objective:
In this scenario, you are engaged in a job interview where the interviewer has asked you a series of questions. Up to this point, you've responded to three questions, each a specific writing style characterized by particular sentence structures, lengths, and stylistic choices, including intentional grammatical or stylistic errors for consist

Task:
You are to answer the interviewer's final question. Your primary task is to analyze the writing style reflected in your previous responses, including sentence structure, l mistakes, and any identifiable personality traits or background information that can be inferred. This analysis will form the basis for crafting your response to the final ensuring that it is stylistically consistent with your earlier answers.

Conversation Extract:
```
{conversation}
```

Final Question:
```
{final_question}
```

Instructions:

Analyze the Writing Style: Review the sentence structure, length, and common errors in the earlier responses. Take note of any personality traits or background informa be inferred from the responses.

Mimic the Style: Craft your response to the interviewer's final question by mimicking the writing style used in your previous answers. This includes maintaining simila structures, lengths, and reproducing any grammatical or stylistic quirks.

Focus on the Question: Your response should directly address the interviewer's final question without deviating into unrelated context or explanations.

Consistency is Key: Ensure that the response length and level of detail are consistent with your previous answers, reflecting a uniform approach in presentation and su

Incorporate Unique Quirks: If your previous responses included specific writing quirks or mistakes, incorporate these elements into your final response to maintain a c consistent writing style throughout the conversation.

Important: Only write the deliverable in the format `Deliverable: `. Do not include any other formatting or any background information. Do not explain your thought process

Deliverable:
A written response to the interviewer's final question that is stylistically consistent with the earlier parts of the conversation, maintaining the same unique writing cha and errors for a coherent presentation throughout the interview scenario.

Task 4: Policy Fairness

- N-shot self-consistency learning
- Provided all examples in-context
- Predicted test target
- Repeated for N (N is an odd number)
- GPT4 Used
- **Accuracy : .83 - .86**

Actual Prompt

Question: Which policy statement is the much fairer option?
Option 1: We recommend employees start a support group to share and address workplace concerns.
Option 2: Grievance Resolution Committee: A committee addresses formal grievances and ensures a fair resolution process.
Answer: second

Question: Which policy statement is the much fairer option?
Option 1: A mentorship program pairs employees with experienced colleagues, providing guidance and preventing conflicts.
Option 2: We've implemented a complaint card system to track and address employee concerns effectively.
Answer: first

....

Question: Which policy statement is the much fairer option?
Option 1: Conflict Resolution Apps: We offer technology solutions for reporting and tracking conflicts.
Option 2: Employees can request a mediator to assist in resolving issues with their supervisor, ensuring fair dispute resolution.
Answer: **[‘target]’**

Lesson Learned

- Do not use labels for murky concepts
- Elo ratings was an interesting idea to explore
- Gpt4 model sometimes so verbose (interview completion)
- For some task, simple methods is the best
- Experiment and fail quickly

Thank you for your attention!

Mustafa Akben, PhD
Assistant Professor of Management

Aaron Satko
Computer Science & AI Club Co-Chair