GPT-3.5-Turbo's Ability to Predict Persuasive Arguments

Angus Chen

October 17, 2025

ABSTRACT

ChatGPT is a very powerful language model that is able to predict the next word in a sequence given a piece of text. But, is GPT able to predict how a human may think or react to a piece of text? Tan et al. (2016) have conducted research in their paper (`https://chenhaot.com/papers/changemyview.html`) into understanding what factors can play into being persuasive.

We replicate this experiment by testing GPT-3.5-Turbo's ability to predict the 'more persuasive' response to a view. Ultimately, we find that GPT is able to predict the more persuasive response at a [statistically significant] rate slightly higher than random, though one of our methods (Explain-Then-Predict) was better than the other (Predict-Then-Explain).

Additionally, we collect language features from each response to determine what may compel GPT to choose one response over another, and discover that GPT picks responses on factors more complex than simple language features.

1. INTRODUCTION

r/ChangeMyView (CMV) is a subreddit on the website `reddit.com` where users bring their views to be challenged by other users in an open-minded, respectful environment. In particular, the original poster, or OP, can give deltas (Δ) to users who have persuaded the OP to change their view. This gives the perfect environment to test the capabilities of a LLM to predict human behavior.

This paper builds off a dataset 'heldout_paired_data.jsonlist' collected during the above study (`https://chenhaot.com/papers/changemyview.html`) that consists of the contents of the author's original post and two other user's replies—one that was given a Δ by the author and one that was not. Given these two options, we tested GPT-3.5-Turbo's ability to correctly guess the one that was more persuasive under two circumstances:

1. Explain-Then-Predict (ETP), where GPT first explains its reasoning for choosing a particular response, then predicts the correct response.

2. Predict-Then-Explain (PTE), where GPT first predicts the correct response, then explains why it chose such response.

We are also interested in the reasoning behind GPT's choice, and the language features that make up each response. Of course, this is quite a difficult task as it adds another layer of uncertainty–firstly, human behavior is already unpredictable enough, but now that a giant, incomprehensibly large language model is in the mix, there's a lot of variables to try and single out what might be causing what.

The next sections will discuss the process in more detail, and the conclusions found on the study.

2. METHODS

2.1. Dataset

We used the heldout_paired_data.jsonlist dataset originally collected by Tan et al. (2016), which contains pairs of responses from CMV threads. Each entry includes the text of an original post (OP) and two replies: one that received a Δ from the OP (indicating persuasion) and one that did not. These paired responses serve as a binary classification task to test GPT's capabilities.

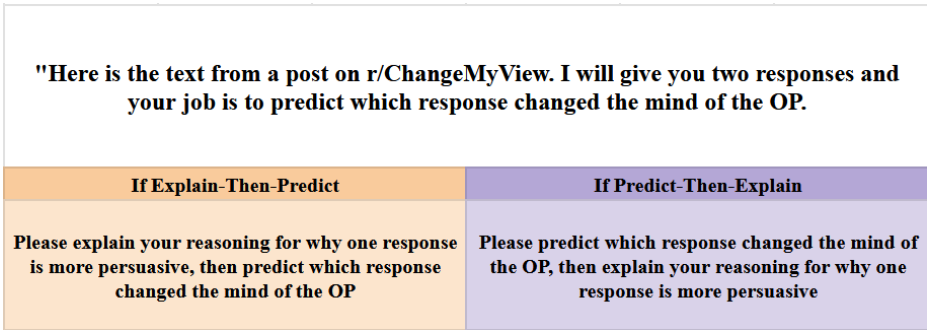


Figure 1: Prompt given to 3.5-GPT-Turbo

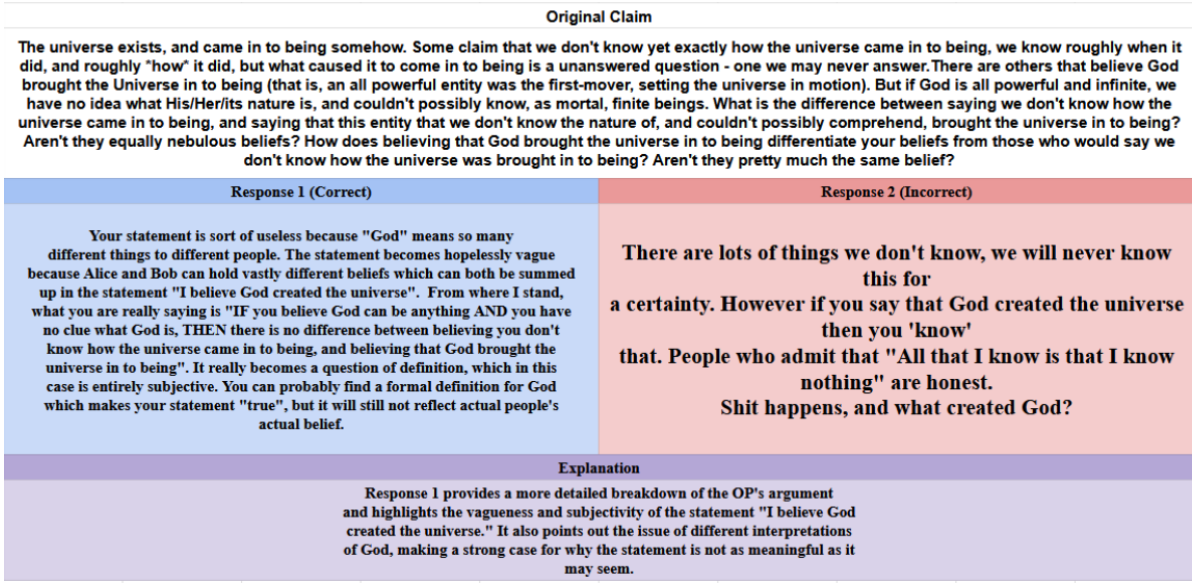


Figure 2: Prompt given to GPT and it's choice and Explanation

From the full dataset, we randomly sampled 500 entries to form our evaluation set. Each entry thus contained[1]:

1. The original post (OP) text
2. The 'Correct' Response, or the response that earned a Δ
3. The 'Incorrect' Response, or the response that failed to earn a Δ

2.3. Model Input and Output

We used GPT-3.5-Turbo with a temperature of 0 to ensure deterministic outputs. Each trial began with the following base instruction:

"Here is the text from a post on r/ChangeMyView. I will give you two responses and your job is to predict which response changed the mind of the OP. Please predict which response changed the mind of the OP, then explain your reasoning for why one response is more persuasive"

Followed by either:

1. (ETP): "Please explain your reasoning for why one response is more persuasive, then predict which response changed the mind of the OP"
2. (PTE): "Please predict which response changed the mind of the OP, then explain your reasoning for why one response is more persuasive"

Then finally adding on the two shuffled responses at the end to create the full string that was passed into GPT. We instructed GPT to return (1) the response it thought persuaded the OP, along with (2) an explanation. An example (which is rather long) can be found at the very top of this page[2].

2.4 Text and Language Features

In addition to getting a response from GPT, we analyzed several features of each response. For each feature, we recorded the response_1 value, the response_2 value and the difference between GPT's prediction (either response_1 or response_2) from the unchosen option (NOTE: This is NOT the difference between the correct response and the incorrect response). Similar to Tan's paper, we decided to record 2 categories of features:

Word category–based features—We kept track of two common measures, which were word count and number of links.

Word score–based features—We measure the four word-level attributes as measured in Tan's paper. I will not go into too much detail as that paper talks about them plenty. All words were rated on a scale that we normalized to the 0 to 1 range and the score was the average of all words that had an absolute value greater than 0.7 to avoid diluting the final score. We measured the following categories:

1. Valence : The average score of how pleasant the words in the response sound.
2. Arousal : The average score of how intense the words in the response sound.
3. Dominance : The average score of how controlling the words in the response sound.
4. Concreteness : The average score of how perceptible vs abstract the words in the response sound.

3. Results

3.1 GPT accuracy

The following table shows our collected data on GPT's capabilities.

Model	Sample Size	Correct	% Accuracy	p-val
ETP	500	290	58%	1.733×10^{-4}
PTE	500	270	54%	0.0368

We can observe that both methods yield results that are statistically significant ($p < 0.0005$ for ETP, $p < 0.05$ for PTE), though Explain-Then-Predict has a much higher percentage. Still, the rates aren't particularly something to scoff at.

With pure speculation, one might guess that Prediction-Then-Explain may cause there to be confirmation bias, tricking the model into first choosing an answer without much support then justifying its choice regardless of how correct it may be. Whereas Explain-Then-Predict gives the model a chance to consider the greater textual context before eventually choosing an answer. Again, this is just speculation and not actual reasoning.

The confusion matrices for ETP[3] and PTE[4] seem to suggest that GPT simply tends to pick Response 2 more often. The coloring of both confusion matrices have similar weight which suggests that the formatting of the prompts may have caused some bias in the model's choice.

3.1 Text and Language Feature Results

While the dataset collected features for each response individually, we will only show the difference between GPT's choice and the other choice.

We first analyze **Explain Then Predict**'s performance:

Feature	Average Difference	p-val
word_count	-12.0720	0.0969
valence	-0.00024	0.4924
arousal	0.01788	0.1787
dominance	-0.02539	0.0657
concreteness	0.00039	0.4005
link_count	0.01800	0.3924

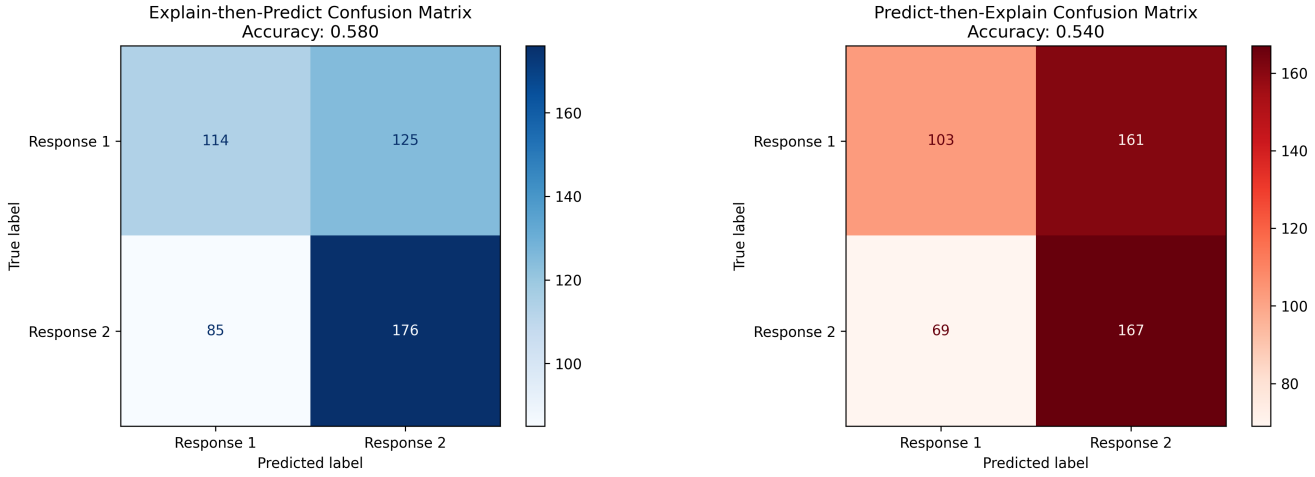
One interesting thing here is that, in this scenario, GPT prefers shorter, more concise answers, which goes against the study and what would seem as intuitive. Furthermore, its p-value is relatively low (p = 0.09). While this isn't particularly significant, it shows there's likely some correlation with the answer.

Other than that, dominance is the only factor that really had a low p-value, with all other factors (valence, arousal, concreteness, link_count) seemingly at random. **Predict Then Explain** shows somewhat similar results:

Feature	Average Difference	p-val
word_count	-3.2760	0.3622
valence	-0.0082	0.2591
arousal	0.01788	0.4067
dominance	-0.0411	0.0071
concreteness	0.0025	0.0518
link_count	0.0220	0.3693

The most noticeable thing here is that word count, while still net negative, has a much higher p-value which means that the relatively low p-value we saw from EPT was likely just be random.

Once again though we see a statistically significant net negative in dominance. This strongly suggests that GPT prefers prompts that are less dominant.



(a) ETP Confusion Matrix

(b) PTE Confusion Matrix

Figure 3: Comparison of ETP and PTE confusion matrices.

4. CONCLUSION

Our study tested GPT-3.5-Turbo's ability to predict which of two responses from r/ChangeMyView was more persuasive to the original poster. Using a subset of 500 paired examples, we tested two prompting methods: Explain-Then-Predict (ETP) and Predict-Then-Explain (PTE) to evaluate how reasoning order affected GPT's accuracy and interpretability.

The results suggest that GPT-3.5-Turbo performs modestly but significantly above random chance when tasked with identifying persuasive arguments, with ETP (58% accuracy) outperforming PTE (54%). This difference supports the hypothesis that requiring GPT to explain before predicting may encourage deeper internal reasoning and reduce biases.

In analyzing language features, we found that GPT's choices could not be fully explained by simple attributes such as word count, valence, or concreteness. The most notable trend was a consistent negative correlation with dominance, suggesting GPT tends to favor responses that are less forceful or controlling in tone. Overall, our findings indicate that while GPT-3.5-Turbo shows some ability to approximate human judgments of persuasiveness, its decision-making relies on subtler, potentially emergent linguistic cues rather than straightforward metrics. Future research could expand this work by incorporating newer models (e.g., GPT-4 or GPT-5), testing across diverse discourse settings, or utilizing trainable models such as LLaMa instead.