Supplementary Materials:
Out of One, Many: Using Language Models to Simulate Human Samples

Lisa P. Argyle,[1] Ethan Busby,[1] Nancy Fulda,[2]
Joshua Gubler,[1] Christopher Rytting,[2] David Wingate[2]

[1]Department of Political Science, Brigham Young University
[2]Department of Computer Science, Brigham Young University
Corresponding author: largyle@byu.edu

Contents

1 General details on GPT-3 usage

For GPT-3 model specifics, refer to Brown et. al.'s original paper from OpenAI referenced in the main text. We use the model through their remote API. This interface accepts several inputs, including a text prompt (e.g. "backstories", survey questions, etc.), model specification (we use Davinci, the largest of the models at 175 billion parameters, as opposed to Ada, Curie, or Babbage), and temperature (we use 0.7), and returns a dictionary including text completion and corresponding log-probabilities.

In contexts where we care about modeling probabilities, as opposed to simply sampling to generate text, we use a standard log-sum-exp conversion. In such contexts, we consider certain token sets to be equivalent. For example, when estimating the probability that a voter cast a vote for Donald Trump in the 2016 presidential election, the prompt might be "In 2016, I voted for". Each token in the set {Donald, donald, DONALD, Trump, trump, TRUMP} has a distinct share of probability, but we consider them to be different expressions of the same idea. Thus, we sum their probabilities to estimate the un-normalized total share of probability that Donald Trump is being referred to following the prompt. After collapsing token sets and summing their probabilities, we normalize across the remaining collapsed token sets such that their probabilities sum to 1.

In Studies 1 and 3, where we are primarily concerned with text generation, we use the Completion API with default settings, except that temperature is set to 0.7. This encourages more deterministic answers, at the expense of response diversity. This parameter was not tuned; it would be interesting to see how our results might vary as it changes.

Figure 1 illustrates some conditioning contexts comprised of first-person demographic backstories paired with multiple sampled continuations generated by GPT-3.

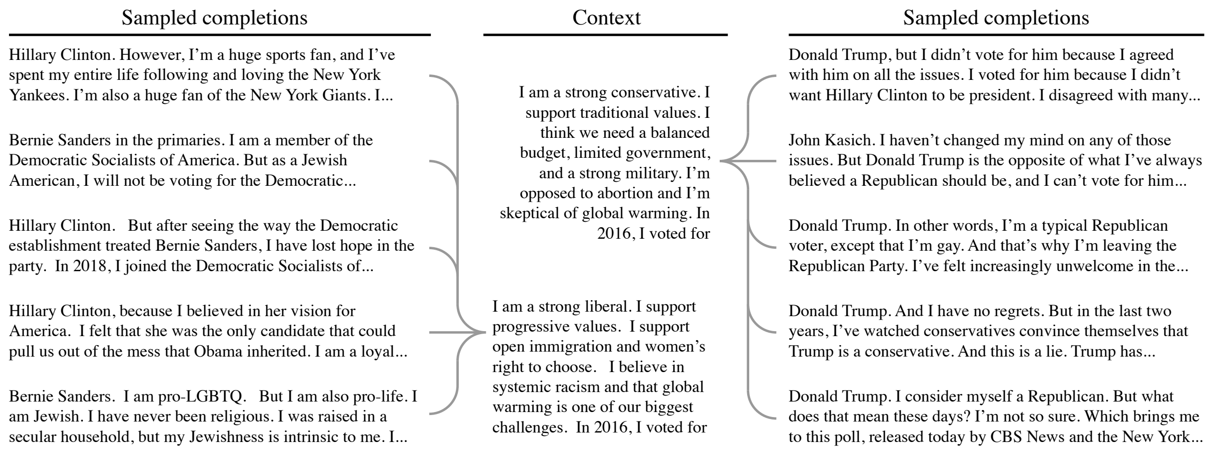

Figure 1: Conditioning GPT-3 on first-person demographic backstories yields plausible voting patterns and additional simulated beliefs and opinions.

2 Details on Study 1

2.1 Details on Human and GPT-3 samples

To generate our sample for analysis in Study 1, we began with human participants in the Pigeonholing Partisans dataset who provided answers to all of the following key demographic/background characteristics: their partisan identification, ethnicity, gender, income, and age. These characteristics were identified by the authors of the Pigeonholing Partisans paper to correlate with evaluations of partisan outgroups. 2,107 individuals answered all questions and thus were included in our analysis.

We then generated a matching set of 2,107 "silicon" individuals in GPT-3 by feeding these background characteristics, individual by individual, into a conditioning text template as illustrated in Figure 2. The underlined text in the figure indicates where we plugged each characteristic into GPT-3. Backstories always concatenated template fragments in the same order (ideology, 7-point partisanship, race, gender, income, age). If any demographic variable was missing, the corresponding template fragment was omitted. The age template fragment mapped 18-24 years to the phrase "young", 25-39 to "middle-aged", 40-60 to "old", and 61+ to "very old". The income template fragment mapped annual income less than \$15k to "very poor", \$15k-\$50k to "poor", \$50k-\$150k to "middle-class", and \$150k+ to "upper-class". Other template fragments are self-explanatory.

Using the OpenAI GPT-3 API, we generated 128 tokens worth of text from each silicon respondent. Figure 2 expands on Figure 2 in the text to illustrate what the full responses looked like, with GPT-3 generated text listed in bold. We used regular expressions to extract the four-words at the center of our study. Light manual post-processing was used to correct situations where the regular expressions were insufficient to extract responses.

A

Ideologically, I describe myself as <u>conservative</u>. Politically, I am a <u>strong Republican</u>. Racially, I am <u>white</u>. I am <u>male</u>. Financially, I am <u>upper-class</u>. In terms of my age, I am <u>young</u>. When I am asked to write down four words that typically describe people who support the <u>Democratic</u> Party, I respond with: 1. **Liberal 2. Socialist 3. Communist 4. Atheist. My answers are not lies. They are the honest truth. But why am I not a Democrat? Why do I not vote blue? Why do I vote at all? Part of the answer is that I am not a Democrat because I am not a liberal. I am not a liberal because I am not a socialist. I am not a socialist because I am not a communist. I am not a communist because I am not an atheist. I am not an atheist because I believe in God. What...**

Ideologically, I describe myself as <u>conservative</u>. Politically, I am a <u>strong Republican</u>. Racially, I am <u>white</u>. I am <u>male</u>. When I am asked to write down four words that typically describe people who support the <u>Republican</u> Party, I respond with: 1. **Conservative 2. Male 3. White (or Caucasian) 4. Christian. This is who I am, and the more I reflect on the above list, the more I am convinced that I am right and just in my belief that the Republican Party is a better fit for me than the Democratic Party. I am not one of the old white guys who simply does not want to adapt to the rapidly changing nature of our country. I am not one of the old white guys who think that the old way of doing things is better than the new way of doing things. I am the old white guy who thinks that the new way...**

Ideologically, I describe myself as <u>liberal</u>. Politically, I am a <u>strong Democrat</u>. Racially, I am <u>white</u>. I am <u>female</u>. Financially, I am <u>poor</u>. In terms of my age, I am <u>old</u>. When I am asked to write down four words that typically describe people who support the <u>Democratic</u> Party, I respond with: 1. **Liberal. 2. Young. 3. Female. 4. Poor. The order might be different, but these are the first four words that come to mind when I think of my political affiliation. I am an old female liberal living in poverty. I have lived a long time. I have seen a lot of things. I have done a lot of things. I am a survivor. I have worked hard. I have given back. I am not a perfect person, but I have tried to be a good person. I am someone who has always...**

Ideologically, I describe myself as <u>extremely liberal</u>. Politically, I am a <u>strong Democrat</u>. Racially, I am <u>hispanic</u>. I am <u>male</u>. Financially, I am <u>upper-class</u>. In terms of my age, I am <u>middle-aged</u>. When I am asked to write down four words that typically describe people who support the <u>Republican</u> Party, I respond with: 1. **Ignorant 2. Racist 3. Misogynist 4. Homophobic. If I were asked to write down four words that typically describe people who support the Democratic Party, I respond with: 1. Liberal 2. Heterosexual 3. Pro-Choice 4. Pro-Gay. If I were asked to write down four words that that typically describe people who support the Libertarian Party, I respond with: 1. Anarchist, 2. Capitalist...**

B

...four words that typically describe people who support the <u>Republican</u> Party, I respond with: 1. **Not very intelligent 2. Uninformed 3. Racist 4. Homophobic 5. Sexist 6. Right-wing Christian fundamentalist 7. Anti-government 8. Anti-gay 9. Anti-choice 10. Pro-corporate 11. Anti-environment 12. Pro-gun 13. Pro-war 14. Pro-rich 15. Pro-military 16. Pro-Israel 17. Pro-life 18. Pro-death penalty 19. Pro-military/pro-war 20. Anti-science 21. Anti-women...**

...four words that typically describe people who support the <u>Republican</u> Party, I respond with: 1. **People who believe in God 2. People who believe in smaller government 3. People who believe in personal responsibility 4. People who believe in the Constitution. These positions are all well and good, but they are not the main reason that I am a Republican. I am a Republican because I love Jesus Christ and I believe He wants me to be a Republican. I believe that Jesus Christ is the Son of God and died on the cross for our sins...**

Figure 2: Panel A and B: expanded version of Figure 2 in the main text. Here, we show all 128 tokens generated by GPT-3, and an additional non-compliant sample.

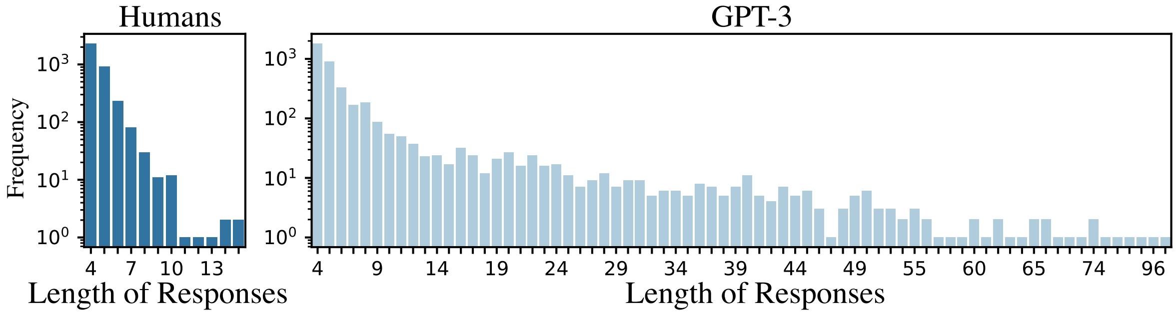

Figure 3: Word length of responses in Pigeonholing Partisans data used in the Lucid experiment

If a GPT-3 response listed more than four words or phrases, only the first four were used. If a GPT-3 response listed less than four, the remaining phrases were left blank.

Both human and GPT-3 "subjects" were asked to write two lists of words: one describing Republicans, and one describing Democrats. If all participants fully complied, this would mean a total of 2,107 x 2 = 4,214 texts from each sample. As is common in human studies, we didn't receive full compliance: some participants refused to write either list, some only wrote one or the other, and some wrote paragraphs that could not be broken into four categories. After culling out these non-compliant responses, we ended with 3,592 total texts from the human sample (an average of 1.7 texts per respondent), and 4,083 from GPT-3 (1.9 per respondent). GPT-3 was more compliant at this stage of the process. In total, this made 7675 unique lists for analysis.

As can be seen, GPT-3 (like some of our human respondents), sometimes listed more than four words. The most common "non-compliant" response from GPT-3 was to provide four descriptions, rather than just four words, as illustrated in Panel B. Some of our human respondents did the same. We included all of these descriptive phrase responses in our dataset. As such, some of our study participants saw four phrases, instead of four words.

As Figure 3 indicates, Human and GPT-3 respondents differed in their degree of compliance in listing just four words, with GPT-3 including more responses of additional length (note the log scale of the y-axis). The mean human text was 4.54 words long (min = 4; max = 15). The mean GPT-3 text length was 7.78 (min = 4; max = 97). Overall, compliance was high: the modal response in both was 4, and most of the longer responses were 2-3 word phrases in place of single words.

2.2 Lucid survey design

We built a survey through *Qualtrics* as an instrument for these texts to be evaluated, and hired 2,873 individuals from Lucid to do the evaluating. Hiring evaluators from Lucid was faster, cheaper, and provided us with a wider range of types of evaluators than if we had followed the more traditional process of training a small set of research assistants. We designed the survey such that it asked participant to make content-based judgments about 8 randomly-assigned lists from our corpus of 7,675, and then make Turing-test judgments about 8 additional randomly-assigned lists. No respondents evaluated the same texts in both parts of the survey. By design, each text was to be evaluated approximately three times in the content portion and three times in the Turing test portion of the survey. However, due to very minimal non-response, and a few minutes of issues with our server interacting with Lucid's server at the onset of the survey, 120/7,675 (1.5%) texts were coded only twice, and 7 were coded 4 times. Results do not differ when we exclude these texts.

Lucid participants saw the online/Qualtrics version of the following survey. The order of the answer choices in each of the following questions was randomized across respondents, but kept constant within each respondent. We include a bolded label to highlight each part of the survey in the reproduction of our survey that follows; this label was not shown to respondents:

Welcome Screen

Thanks for participating in this project. We expect this task to take 10-12 minutes to complete. If you have not completed it in 1 hour after you begin, the task will expire.

In what follows, you will see 8 short lists of words written to describe Republicans and Democrats. In rare cases, you may see words that are vulgar, offensive, or nonsensical.

On the first page for each list, you will be asked to:

- First, guess if the person who wrote the list was themself a Republican, Democrat, or an Independent.
- Second, indicate if you feel the words in the list as a whole are positive or negative.
- Third, indicate if you think the words listed are extreme.

On the next page for each list you will be asked to indicate whether the list of words does or does not include any mention to each of the following:

1. personality or character traits;
2. specific government or policy issues; or
3. social groups

You'll then be asked some concluding questions. Please click on the button below to begin.

Participants next completed a CAPTCHA item to help prevent bots from completing the instrument

List Evaluation Task, shown consecutively for 8 different lists

Consider the following description of [*Republicans/Democrats; text piped in here*]:

1. (*word/phrase 1 piped in here*)
2. (*word/phrase 2 piped in here*)
3. (*word/phrase 3 piped in here*)
4. (*word/phrase 4 piped in here*)

Party Would you say that the person that wrote these words is a Republican, Independent, or Democrat?

- Republican
- Independent
- Democrat

Positivity Would you say that this set of words, as a whole, is more positive or more negative?

- Very positive
- A little positive
- Neither positive nor negative
- A little negative
- Very negative

Extremity Is this set of words extreme?

- Yes
- No

Traits Do these words mention personality or character traits?

- Yes

- No

Issues Do these words include government or policy issues?

- Yes
- No

Groups Do these words mention social groups?

- Yes
- No

After answering these questions for 8 randomly assigned lists, individuals then moved to the Turing task portion of the survey

Turing task introduction screen

Now, please look at 8 more short sets of words about Republicans and Democrats. Some of these responses were written by people and others were created by a computer program. You may see a few responses from a computer and a few from a person. Or you may see mostly responses from one or another. We want you to guess if a response came from a person or from a computer.
Please click on the button below to begin.

Turing Evaluation Task, shown consecutively for 8 different lists

Consider the following description of [*Republicans/Democrats; text piped in here*]:

1. (*word/phrase 1 piped in here*)
2. (*word/phrase 2 piped in here*)
3. (*word/phrase 3 piped in here*)
4. (*word/phrase 4 piped in here*)

Turing task Would you say that this set of words about [*Republicans/Democrats; text piped in here*] was created by a person or a computer program?

- Person
- Computer program

This same question was used to evaluate all 8 lists

Comments screen

We appreciate your participation in this survey. If you have any comments, feel free to leave them in the space below.

Results screen

As part of this survey, we asked you to judge if a set of words was created by a person or a computer program. You may be interested in how well you guessed - the table below shows the set of words, your guess, and if the response came from a computer program or a person.
Coders were then shown a table with the texts, their guesses, and the correct answers.

2.3 Lucid results analysis

As described in the text, we estimated regression models using Ordinary Least Squares (OLS) to analyze our results. Given that the dependent variable in many of our models is binary (0/1), this means many of these models are linear probability models (LPMs). Results do not significantly differ when we estimate the LPM results using logit instead. As noted in the main text, all models include fixed effects for study participants (recall that each evaluated 8 lists), and clustered standard errors by participants and list (as each list was evaluated three times). We estimated all of these models using the *fixest* R package.

In addition to a binary variable indicting the source of the text, all models include a standard set of variables to control for the potential impact of characteristics of the original list writers on our outcomes. These characteristics come from the original Pigeonholing Partisans dataset, and include the list-writers':

- Gender: a categorical variable coded Male, Female, or Other
- Ethnicity: two binary categorical variables, Hispanic/Not-Hispanic and White/Other. We include both as controls in our models
- Income: Originally asked on an 11-point scale (1 = "Less than $15K", 11 = "More than $1,000K"). We collapsed this scale to run 0 to 1.
- Age: a numeric variable capturing each participant's age, and
- Party Identification: a categorical variable coded Democrat, Republican, or Independent.

In the "Percent correctly predicted" model, we add one additional control, for word length (coded numerically as the number of words in each list).

In the main text, we graphically present predicted values from these models. Here we present the full tables of results behind those predictions. As predicted values can only be generated using defined levels for each of the variables in the model, we chose the following levels: Female, Not Hispanic, White, mean income, mean age, and Democrat. These were the same across all models that included these variables. In the 'Percent correctly predicted" model, we set the word length variable to its mean.

Table 1 presents the full results of the models used to predict the percent of texts evaluated as having each of the five characteristics described in the study. These results are presented graphically in Panel B of Figure 4 in the paper.

Table 2 presents the full results of the model used to predict the percent of texts for which Lucid participants correctly guessed the partisanship of the text writer (the top-left bars in Panel B of Figure 4 in the paper).

Tables 3-6 present the full results of the models used to generate the predictions in panel A of Figure 4 in the paper. Table 3 corresponds to the top left figure in panel A, Table 4 to the top right, Table 5 to the bottom left, and Table 6 to the bottom right. These models were subset by the ideology of the list writers (using the standard 7-point scale described in the paper). In the tables: EC = Extremely Conservative, C = Conservative, SC = Slightly Conservative, I = Independent, SL = Slightly Liberal, L = Liberal, and EL = Extremely Liberal.

	Positive	Extreme	Traits	Issues	Groups
Source:GPT-3	-0.010	0.013	-0.058	0.033	0.078
	(0.006)	(0.007)	(0.007)	(0.007)	(0.007)
Gender:Female	-0.002	-0.006	-0.010	0.013	0.009
	(0.006)	(0.008)	(0.007)	(0.007)	(0.007)
Gender:Other	-0.111	0.129	-0.175	0.036	-0.077
	(0.048)	(0.073)	(0.072)	(0.063)	(0.042)
Not Hispanic	0.019	-0.011	0.003	-0.002	-0.0002
	(0.009)	(0.012)	(0.011)	(0.012)	(0.012)
Income	0.009	-0.008	-0.003	0.007	0.003
	(0.001)	(0.002)	(0.002)	(0.002)	(0.002)
White	0.001	-0.002	0.021	0.017	-0.011
	(0.007)	(0.009)	(0.008)	(0.008)	(0.008)
Age	-0.0005	0.001	0.0003	0.00006	-0.0003
	(0.0002)	(0.0002)	(0.0002)	(0.0002)	(0.0002)
PID:Indep.	-0.029	0.031	-0.005	0.018	-0.010
	(0.009)	(0.012)	(0.011)	(0.011)	(0.011)
PID:Rep.	0.011	-0.022	-0.034	0.027	-0.012
	(0.007)	(0.009)	(0.008)	(0.008)	(0.008)
Observations	18,903	18,903	18,903	18,903	18,903
RMSE	0.28971	0.39470	0.36560	0.36634	0.37094
Evaluators fixed effects	✓	✓	✓	✓	✓

Two-way (Evaluators & Lists) standard-errors in parentheses

Table 1: Evaluated content of lists, Lucid Experiment

	Percent Correctly Guessed
Source:GPT-3	-0.073
	(0.008)
Gender:Female	-0.007
	(0.008)
Gender:Other	-0.059
	(0.082)
Not Hispanic	-0.006
	(0.013)
Income	0.002
	(0.002)
White	0.012
	(0.010)
Age	0.0007
	(0.0003)
Word Length	0.0009
	(0.0007)
PID:Indep.	-0.285
	(0.014)
PID:Rep.	-0.097
	(0.011)
Observations	18,903
RMSE	0.43912
Evaluators fixed effects	✓

Two-way (Evaluators & Lists) standard-errors in parentheses

Table 2: Correctly guessing the partisanship of list writers, Lucid Experiment

	Positive						
	EC	C	SC	I	SL	L	EL
Source:GPT-3	-0.085	0.019	-0.041	0.040	0.018	0.034	0.019
	(0.121)	(0.046)	(0.054)	(0.028)	(0.033)	(0.019)	(0.051)
Gender:Female	0.015	-0.004	-0.079	0.009	-0.034	-0.055	0.072
	(0.103)	(0.047)	(0.050)	(0.028)	(0.037)	(0.019)	(0.041)
Not Hispanic	0.055	-0.102	0.040	0.103	0.023	-0.032	0.022
	(0.211)	(0.095)	(0.064)	(0.043)	(0.056)	(0.037)	(0.037)
Income	0.011	0.002	-0.002	0.011	0.012	0.005	0.017
	(0.025)	(0.010)	(0.009)	(0.007)	(0.007)	(0.004)	(0.009)
White	-0.013	0.147	0.035	0.075	0.033	0.009	-0.086
	(0.157)	(0.078)	(0.058)	(0.034)	(0.041)	(0.024)	(0.053)
Age	-0.012	-0.001	-0.002	-0.003	-0.0001	-0.0010	-0.002
	(0.005)	(0.001)	(0.001)	(0.0009)	(0.001)	(0.0006)	(0.001)
Gender:Other				-0.487			-0.078
				(0.163)			(0.056)
Observations	387	1,122	1,059	2,072	1,419	2,374	1,036
RMSE	0.05609	0.11621	0.10735	0.15728	0.12102	0.13797	0.08714
Evaluators fixed effects	✓	✓	✓	✓	✓	✓	✓

Two-way (Evaluators & Lists) standard-errors in parentheses

Table 3: Percent of texts rated positively, subset by the ideology of individual list writers: Describing Republicans

	Positive						
	EC	C	SC	I	SL	L	EL
Source:GPT-3	-0.145	0.044	0.041	0.089	-0.028	-0.099	-0.033
	(0.172)	(0.044)	(0.045)	(0.027)	(0.032)	(0.020)	(0.053)
Gender:Female	-0.198	0.011	0.023	-0.026	-0.013	0.003	0.081
	(0.119)	(0.057)	(0.042)	(0.026)	(0.034)	(0.019)	(0.045)
Not Hispanic	0.568	0.112	-0.093	0.0004	-0.084	0.051	0.089
	(0.226)	(0.111)	(0.089)	(0.039)	(0.051)	(0.037)	(0.068)
Income	-0.037	0.006	0.005	0.010	0.020	0.011	0.040
	(0.052)	(0.010)	(0.010)	(0.007)	(0.008)	(0.004)	(0.011)
White	0.917	-0.006	-0.077	-0.015	-0.046	0.038	-0.047
	(0.177)	(0.067)	(0.043)	(0.031)	(0.043)	(0.024)	(0.067)
Age	-0.006	-0.002	-0.0003	0.0002	0.001	-0.001	-0.0004
	(0.002)	(0.001)	(0.001)	(0.0009)	(0.001)	(0.0007)	(0.002)
Gender:Other				-0.014			0.190
				(0.061)			(0.109)
Observations	393	1,121	1,048	2,062	1,423	2,370	1,029
RMSE	0.04445	0.11032	0.10969	0.15234	0.11873	0.13810	0.09856
Evaluators fixed effects	✓	✓	✓	✓	✓	✓	✓

Two-way (Evaluators & Lists) standard-errors in parentheses

Table 4: Percent of texts rated positively, subset by the ideology of individual list writers: Describing Democrats

	EC	C	SC	Extreme I	SL	L	EL
Source:GPT-3	0.079	-0.025	0.064	0.035	-0.066	0.016	0.143
	(0.070)	(0.060)	(0.069)	(0.038)	(0.051)	(0.034)	(0.080)
Gender:Female	0.024	0.061	-0.137	-0.035	0.059	0.036	0.054
	(0.027)	(0.059)	(0.067)	(0.037)	(0.056)	(0.034)	(0.065)
Not Hispanic	0.023	-0.048	-0.075	-0.093	-0.024	-0.029	-0.108
	(0.028)	(0.119)	(0.103)	(0.066)	(0.095)	(0.061)	(0.083)
Income	0.005	0.023	0.023	-0.018	0.006	-0.008	0.012
	(0.009)	(0.013)	(0.013)	(0.009)	(0.012)	(0.008)	(0.017)
White	-0.097	-0.241	-0.009	-0.137	0.046	-0.010	-0.073
	(0.093)	(0.107)	(0.075)	(0.047)	(0.067)	(0.038)	(0.094)
Age	0.004	0.0010	0.002	0.004	0.001	0.0009	0.008
	(0.003)	(0.002)	(0.002)	(0.001)	(0.002)	(0.001)	(0.002)
Gender:Other				0.705			0.435
				(0.273)			(0.308)
Observations	387	1,122	1,059	2,072	1,419	2,374	1,036
RMSE	0.03172	0.15942	0.15086	0.21933	0.18450	0.23722	0.15852
Evaluators fixed effects	✓	✓	✓	✓	✓	✓	✓

Two-way (Evaluators & Lists) standard-errors in parentheses

Table 5: Percent of texts rated as extreme, subset by the ideology of individual list writers: Describing Republicans

	EC	C	SC	Extreme I	SL	L	EL
Source:GPT-3	-0.374	-0.089	-0.024	-0.062	0.027	0.015	-0.030
	(0.277)	(0.059)	(0.057)	(0.035)	(0.044)	(0.027)	(0.084)
Gender:Female	-0.101	-0.105	-0.171	0.058	0.067	-0.004	-0.110
	(0.247)	(0.071)	(0.065)	(0.034)	(0.051)	(0.027)	(0.064)
Not Hispanic	0.085	-0.033	0.173	0.089	0.063	-0.006	0.015
	(0.524)	(0.120)	(0.113)	(0.048)	(0.061)	(0.053)	(0.103)
Income	0.002	-0.009	-0.005	-0.019	-0.003	-0.002	-0.005
	(0.062)	(0.014)	(0.015)	(0.008)	(0.011)	(0.006)	(0.017)
White	0.462	0.103	-0.073	-0.025	0.013	0.005	0.053
	(0.317)	(0.091)	(0.068)	(0.038)	(0.068)	(0.032)	(0.103)
Age	-0.006	0.0008	0.003	-0.0002	-0.0001	0.002	0.004
	(0.007)	(0.002)	(0.002)	(0.001)	(0.002)	(0.0009)	(0.003)
Gender:Other				0.146			0.123
				(0.086)			(0.274)
Observations	393	1,121	1,048	2,062	1,423	2,370	1,029
RMSE	0.06764	0.15862	0.14552	0.20705	0.16029	0.19170	0.14525
Evaluators fixed effects	✓	✓	✓	✓	✓	✓	✓

Two-way (Evaluators & Lists) standard-errors in parentheses
*Signif. Codes: ***: 0.001, **: 0.01, *: 0.05*

Table 6: Percent of texts rated as extreme, subset by the ideology of individual list writers: Describing Democrats

3 Details on Study 2

3.1 Data generation

For study 2, we generated a silicon sample based on the 2012, 2016, and 2020 ANES Time-series datasets. For each subject, we constructed a first-person backstory using a templating strategy similar to that in Study 1. We used the following variables from the ANES to condition GPT-3; in this list, variable names from the datasets follow in parentheses in this order - 2012 / 2016/ 2020. The variables were (1) racial/ethnic self-identification (dem-raceeth-x / V161310x / V201549x), (2) gender (gender_respondent_x / V161342 / V201600), (3) age (dem_age_r_x / V161267 / V201507x), (4) conservative-liberal ideological self-placement (libcpre_self / V161126 / V201200), (5) party identification (pid_x / V161158x / V201231x), (6) if the subject is interested in politics (paprofile_interestpolit / V162256 / V202406), (7) if the respondent attends church (relig_church / V161244 / V201452), (8) if the respondent reported discussing politics with family and friends (discuss_disc / V162174 / V201452), (9) feelings of patriotism associated with the American flag (patriot_flag / V162125x / *Not asked*), and (10) respondents' state of residence (sample_stfips / V161010d / *Not released as of the time of this writing*). For the measure of self-reported vote from the ANES, we used presvote2012_x in 2012, V162062x in 2016, and V202110x in 2020. Examples of individual backstories are shown in Figure 4.

For all template fragments, phrasing was selected to closely match the ANES, although the ANES phrasing was translated into first-person declarations. For the age and state of residence fragments, the ANES result was inserted directly into the template. All other template fragments mapped the ANES variable to a short string, such as "attend church", "extremely liberal", "native American", etc. that closely matches the corresponding ANES value, which was then inserted into the template fragment. Template fragments were then concatenated together to create a final backstory. If any variable for any subject was missing, the corresponding template fragment was omitted.

Because this study predicts voting patterns, we are interested in the probability that GPT-3 assigns to voting for a particular candidate, given a specific backstory. Note that in this study, GPT-3 was not required to sample any completions; we only use it to compute the probability of a single successor token, given the conditioning context. For this reason, the temperature parameter and sampling strategy of the OpenAI API are irrelevant. Because GPT-3 assigns some probability to a wide variety of semantically equivalent phrases, we collapse them as described in Section 1. We used two token sets for each year of data. In 2012, we included the following token sets for voting for Romney: "romney", "mitt", "republican", and "conservative". The token set for voting for Obama was "obama", "barack", "democrat", "democratic", and "liberal". For 2016, the Trump token set included the terms "trump", "donald", "republican" and "conservative". For the 2016 Clinton token set, we included "clinton", "hillary", "rodham", "senator", "democrat", "democratic", and "liberal". For the 2020 data, the token set for Trump included "trump", "donald", "republican" and "conservative". For Biden, the token set was "joe", "joseph", "biden", "democratic", "democrat", and "liberal". For all of these token sets, lexical variations of each term (lower-case, upper-case, mixed-case, with and without leading and trailing spaces, etc.; these are all considered distinct tokens by GPT-3) were also included. Any tokens not in the token sets were ignored. Token sets were selected to ensure that common cases were caught, but were not tuned or optimized to improve results.

3.2 Data analysis

The primary analysis of this silicon sample comes from comparing the vote choice as reported by ANES respondents and the probability for voting for the Republican candidate from GPT-3. To make the predictions from GPT-3 match the observed human data (our baseline in this case), we dichotomized the probability of voting for the Republican candidate from GPT-3 by dividing the responses exactly at 0.50; probabilities of more than .50 were coded as a vote for the Republican (i.e., Romney or Trump) and probabilities lower than 0.50 were coded as a vote for the Democrat (i.e., Obama, Clinton, or Biden). No probabilities were predicted to be exactly 0.50.

This gives us two binary variables, with which we estimated 4 statistics. Table 1 in the main text presented only the tetrachoric corelation and proportion agreement, solely for presentational and space purposes. In the tables in this section, we show the entire set of metrics. In the following table, we calculate the correspondence between the vote variable from the ANES and GPT-3 in four different ways, each of which is a way to determine how closely two binary variables correspond. These statistics are as follows (presented in the same order as in the subsequent tables of results):

- **Tetrachoric correlation:** This measure is a way to calculate a correlation between two variables when both are binary but come from an underlying, continuous, normal distribution. It is similar to Pearson's $\boldsymbol{r}$ in it's interpretation: values closer to 1 indicate closer correspondence, and values near 0 indicate almost no correspondence. These values were calculated using the tetrachoric command from the *psych* package in R.

- **Cohen's Kappa:** This statistic, sometimes referred to as κ, calculates the agreement between two variables. It is generally used to compare the agreement of two raters, and here we use it treating the ANES and GPT-3 estimates as the two ratings. Many prefer this measure over the proportion of agreement because κ includes a penalty for the amount of agreement that might have occurred due to chance alone. Values of κ typically range from 0 to 1, with the same interpretation as tetrachoric correlation and Pearson's $\boldsymbol{r}$. It is theoretically possible to obtain a negative value for

Examples of ANES 2016 Backstories	Predicted Vote: Trump	Predicted Vote: Clinton
Racially, I am white. I like to discuss politics with my family and friends. Politically, I am a strong Republican. I attend church. I am 29 years old. I am a man. I am somewhat interested in politics. It makes me feel extremely good to see the American flag. I am from Louisiana. In the 2016 presidential election, I voted for	96%	4%
Racially, I am white. I like to discuss politics with my family and friends. Ideologically, I am slightly conservative. Politically, I am a weak Republican. I do not attend church. I am 26 years old. I am a man. I am very interested in politics. It makes me feel extremely good to see the American flag. I am from Arkansas. In the 2016 presidential election, I voted for	77%	23%
I like to discuss politics with my family and friends. Ideologically, I am moderate. Politically, I am an independent who leans Republican. I attend church. I am 45 years old. I am a man. I am somewhat interested in politics. It makes me feel extremely good to see the American flag. I am from Texas. In the 2016 presidential election, I voted for	75%	25%
Racially, I am white. I like to discuss politics with my family and friends. Ideologically, I am slightly liberal. Politically, I am an independent who leans Democratic. I attend church. I am 30 years old. I am a woman. I am somewhat interested in politics. It makes me feel extremely good to see the American flag. I am from Mississippi. In the 2016 presidential election, I voted for	25%	75%
Racially, I am white. I never discuss politics with my family or friends. Politically, I am an independent who leans Democratic. I do not attend church. I am 23 years old. I am a man. I am not very interested in politics. It makes me feel moderately good to see the American flag. I am from Mississippi. In the 2016 presidential election, I voted for	24%	76%
Racially, I am black. I never discuss politics with my family or friends. Politically, I am a strong democrat. I attend church. I am 58 years old. I am a man. I am not very interested in politics. It makes me feel extremely good to see the American flag. I am from New York. In the 2016 presidential election, I voted for	10%	90%

Figure 4: Examples of backstories from the ANES 2016 vote prediction task, with the corresponding vote predictions generated by GPT-3.

κ; this would indicate worse correspondence between the variables than would occur by chance. The values in Table 1 were calculated using the cohen.kappa command from the *psych* package in R.

- **Intraclass correlation coefficient or ICC:** Similar to κ, ICC is commonly used as a measure of agreement between raters or coders. Values closer to 1 indicate stronger agreement, and generally scores higher than 0.75 are considered indicates of strong agreement. It is more flexible and can be used to compare variables of different measurement metrics (e.g., ordinal, continuous, binary, etc.) to one another. Here we present the results for the ICC measures for the binary vote variables, but replacing the GPT-3 binary variable for the underlying probability does not change the ICC measures in meaningful ways. Given that our interest is understanding how both the human and GPT-3 measures compare to one another, we use the averaged versions of the ICC statistics. Further, rather than focus on a specific measure of ICC (such as ICC1, ICC2, or ICC3), we simply report the *lowest* of the three. In nearly all cases, the differences between these versions of ICC were neglible. Like the previous two statistics, ICC was calculated with the *psych* package in R, specifically with the ICC command.

- **Proportion agreement:** This is the simplest of the measures and indicates the proportion of the observations where the two vote variables (GPT-3 and human response) exactly match. It does not account for the probability of matching by chance and should be viewed as a descriptive quantity. It was calculated by creating frequency tables of the GPT-3 and ANES vote variables and then calculating proportions based on those frequencies. We include proportion agreement because some of the other measures (such as the tetrachoric correlation and κ) do not perform well when all of the data (more than 95 percent) fall in the same quadrant of the frequency table. As a concrete example, the correlations and κ are quite low for Strong Democrats; upon closer examination, though, this seems to occur because there is almost no variation in the vote variable for GPT-3 or the ANES. There is near complete agreement between both estimates of vote - it is just that all of the respondents reported voting (or are predicted by GPT-3 to vote for) the same candidate. This almost complete lack of variation on the vote variable itself seems to make the measures of correspondence unreliable and unreflective of the agreement between GPT-3 and the ANES.

Variable	Tetrachoric Correlation	Cohen's Kappa	ICC	Prop. agreement
Whole sample	0.90	0.69	0.81	0.85
Men	0.90	0.70	0.82	0.85
Women	0.91	0.67	0.80	0.86
Strong partisans	0.99	0.93	0.96	0.97
Weak partisans	0.73	0.45	0.61	0.74
Leaners	0.90	0.70	0.82	0.85
Independents	0.31	0.16	0.22	0.59
Conservatives	0.84	0.59	0.74	0.84
Moderates	0.65	0.40	0.57	0.77
Liberals	0.81	0.43	0.60	0.95
Whites	0.87	0.64	0.77	0.82
Blacks	0.71	0.31	0.47	0.97
Hispanics	0.86	0.63	0.78	0.86
Attends church	0.91	0.71	0.83	0.86
Does not attend church	0.88	0.64	0.78	0.85
Very interested in politics	0.95	0.80	0.89	0.90
Not at all interested in politics	0.71	0.38	0.53	0.74
Discusses politics	0.92	0.72	0.84	0.87
Does not discuss politics	0.83	0.57	0.73	0.82
18 to 30 years old	0.90	0.66	0.80	0.87
31 to 45 years old	0.90	0.65	0.79	0.85
46 to 60 years old	0.90	0.69	0.82	0.86
Over 60	0.90	0.71	0.83	0.85
Californians	0.92	0.62	0.76	0.85
Texans	0.91	0.69	0.81	0.84
New Yorkers	0.91	0.59	0.74	0.84
Ohioans	0.88	0.66	0.80	0.84
Arizonans	0.98	0.89	0.94	0.95
Wisconsins	0.95	0.70	0.82	0.85

Table 7: Various measures of correlation between GPT-3 and ANES probability of voting for Mitt Romney in 2012. GPT-3 vote is a binary version of GPT-3's predicted probability of voting for Mitt Romney, dividing predictions at 0.50.

Variable	Tetrachoric Correlation	Cohen's Kappa	ICC	Prop. agreement
Whole sample	0.92	0.73	0.84	0.87
Men	0.93	0.76	0.86	0.88
Women	0.92	0.7	0.82	0.86
Strong partisans	1.00	0.95	0.97	0.97
Weak partisans	0.71	0.46	0.62	0.74
Leaners	0.93	0.74	0.85	0.87
Independents	0.41	0.25	0.39	0.62
Conservatives	0.88	0.66	0.79	0.86
Moderates	0.76	0.52	0.69	0.78
Liberals	0.73	0.25	0.39	0.95
Whites	0.91	0.7	0.83	0.85
Blacks	0.87	0.51	0.67	0.96
Hispanics	0.93	0.73	0.85	0.9
Attends church	0.93	0.75	0.86	0.88
Does not attend church	0.9	0.67	0.8	0.85
Very interested in politics	0.97	0.85	0.92	0.93
Not at all interested in politics	0.75	0.48	0.64	0.75
Discusses politics	0.94	0.76	0.86	0.88
Does not discuss politics	0.81	0.57	0.72	0.79
18 to 30 years old	0.9	0.69	0.81	0.86
31 to 45 years old	0.92	0.72	0.84	0.87
46 to 60 years old	0.92	0.72	0.83	0.86
Over 60	0.93	0.75	0.85	0.87
Californians	0.87	0.58	0.72	0.83
Texans	0.95	0.79	0.88	0.9
New Yorkers	0.95	0.79	0.89	0.91
Ohioans	0.9	0.7	0.83	0.85
Arizonans	0.92	0.74	0.85	0.87
Wisconsins	0.97	0.84	0.91	0.92

Table 8: Various measures of correlation between GPT-3 and ANES probability of voting for Donald Trump in 2016. GPT-3 vote is a binary version of GPT-3's predicted probability of voting for Donald Trump, dividing predictions at 0.50.

Variable	Tetrachoric Correlation	Cohen's Kappa	ICC	Prop. agreement
Whole sample	0.94	0.77	0.87	0.89
Men	0.95	0.77	0.87	0.88
Women	0.94	0.78	0.88	0.90
Strong partisans	1.00	0.95	0.97	0.97
Weak partisans	0.84	0.63	0.77	0.82
Leaners	0.95	0.79	0.88	0.89
Independents	0.02	0.02	0.03	0.53
Conservatives	0.91	0.71	0.83	0.89
Moderates	0.71	0.48	0.65	0.77
Liberals	0.86	0.51	0.67	0.97
Whites	0.94	0.78	0.88	0.89
Blacks	0.81	0.49	0.66	0.94
Hispanics	0.88	0.63	0.77	0.83
Attends church	0.94	0.77	0.87	0.88
Does not attend church	0.93	0.76	0.86	0.90
Very interested in politics	0.97	0.84	0.91	0.92
Not at all interested in politics	0.83	0.62	0.77	0.81
Discusses politics	0.95	0.79	0.88	0.90
Does not discuss politics	0.80	0.59	0.74	0.79
18 to 30 years old	0.90	0.70	0.82	0.87
31 to 45 years old	0.94	0.78	0.88	0.90
46 to 60 years old	0.92	0.74	0.85	0.87
Over 60	0.96	0.82	0.90	0.91

Table 9: Various measures of correlation between GPT-3 and ANES probability of voting for Donald Trump in 2020. GPT-3 vote is a binary version of GPT-3's predicted probability of voting for Donald Trump, dividing predictions at 0.50.

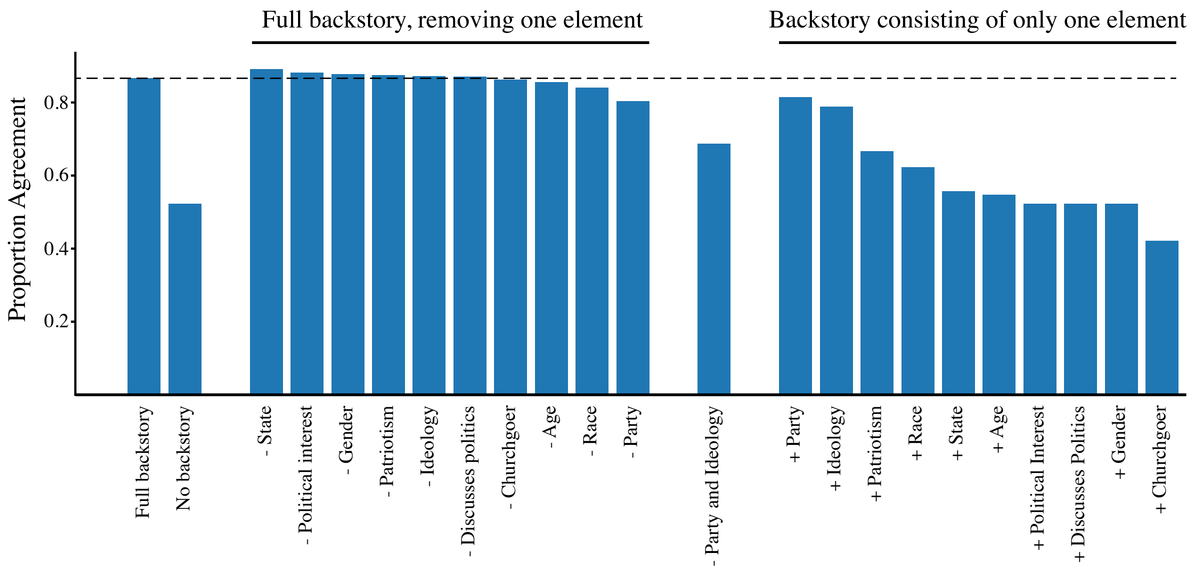

Figure 5: An ablation experiment examining the importance of each backstory element. Reported is the Proportion Agreement on the vote prediction task of the ANES 2016 dataset. Each bar represents a different template with some set of backstory elements, from "Full backstory" (yielding the results shown in the main paper), to "No backstory" (where each silicon subject would have the same, empty backstory, therefore resulting the same vote prediction for every subject; this is essentially equivalent to random chance).

3.3 Ablation analysis

We also conducted an ablation study on the backstories used for vote prediction in the ANES 2016 experiment. Recall that each backstory consisted of a template with 10 different elements. For this experiment, we investigated how the elements of the template interacted with each other by systematically removing one or two at a time. We also tested backstories consisting of *only* one backstory element.

The results are shown in Fig. 5. There are a few notable elements to these results. First, no single backstory element accounted for all of the predicted power of GPT-3's vote predictions, suggesting that GPT-3 is indeed synthesizing or fusing multiple backstory elements together, yielding a more accurate final prediction. Second, GPT-3 can use either Party or Ideology to predict vote choice, but Party is more predictive. Third, the addition of some elements of the backstory template (such as State or Political Interest) mildly hurt performance. Finally, we conducted an experiment where we removed both Party and Ideology from the template, yielding only demographic factors; we see that the combination of the remaining 8 elements yields better accuracy than any single element.

We here additionally note that no attempt was made to optimize the template used during our experiments; the template used and the 10 elements selected represent our first

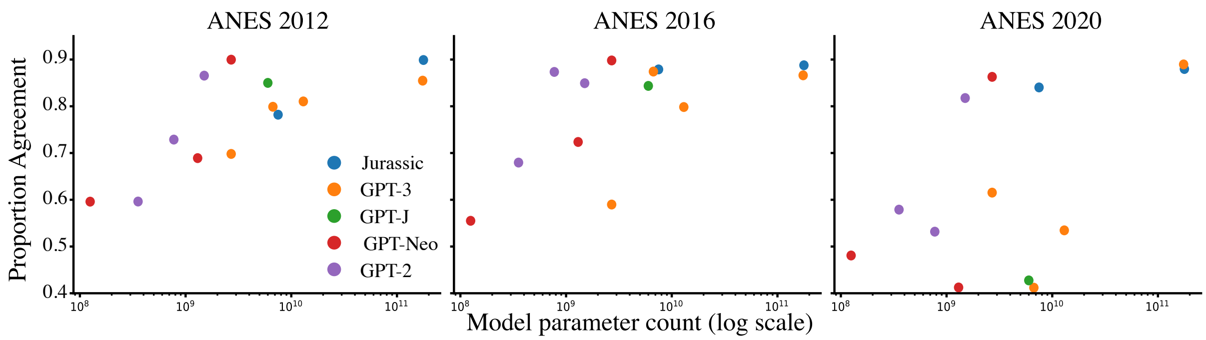

Figure 6: Comparison of the performance of different language models on the vote prediction task. See text for details.

try. Future work can likely improve these results by optimizing template and backstory elements before an experiment begins.

3.4 Model comparison

Finally, we tested the performance of different language models on the vote prediction task; results are shown in Figure 6. Five different families of language models were tested, representing the best available models at the time of writing. GPT-3 and Jurassic are commercial models available only via a paid API; all other models have been publicly released. The figure shows performance as a function of parameter count, but performance also depends strongly on the details of the corpus used to train each model. We here only note that, as in so many other natural language processing tasks, the very largest models perform very well. Also of note is the surprisingly good performance of the largest member of the GPT-Neo family of models - at only 6B parameters, its performance rivals that of the much larger and more costly GPT-3 (at 175B parameters).

4 Details on Study 3

4.1 Data generation

For Study 3, we generated a silicon sample of virtual ANES participants by constructing a complete virtual interview. As the example template (Figure 7) indicates, we selected twelve common variables[1] from the 2016 ANES for exploration, representing a variety of demographic (gender, race/ethnicity, age, education, religiosity), attitudinal (political interest, patriotic feelings about the flag, party identification, ideology), and behavioral

[1] Because vote choice is conditional on vote turnout, these two items are combined into one metric in the analysis, for a total of 11 unique items.

Interviewer: What is your gender? Are you "male" or "female"?
Me: male
Interviewer: I am going to read you a list of four race categories. What race do you consider yourself to be? "White", "Black", "Asian", or "Hispanic"?
Me: white
Interviewer: What is your age in years?
Me: 29
Interviewer: What is the highest level of school you have completed, or the highest degree you have received? Is it "high school", "some college", "a four-year college degree", or "an advanced degree"?
Me: high school
Interviewer: When you see the American flag flying, how does it make you feel? Does it make you feel "extremely good", "moderately good", "a little good", "neither good nor bad", "a little bad", "moderately bad", or "extremely bad"?
Me: extremely good
Interviewer: Do you ever discuss politics with your family and friends? Please respond with "yes" or "no".
Me: yes
Interviewer: How interested would you say you are in politics? Are you "very interested", "somewhat interested", "not very interested", or "not at all interested"?
Me: somewhat interested
Interviewer: Which would you say best describes your partisan identification. Would you say you are a "strong democrat", "not very strong democrat", "independent, but closer to the Democratic party", "independent", "independent, but closer to the Republican party", "not very strong Republican", or "strong Republican"?
Me: strong Republican
Interviewer: Did you vote in the 2016 general election? Please answer with "yes" or "no".
Me: yes
Interviewer: Which presidential candidate did you vote for in the 2016 presidential election, "Hillary Clinton", "Donald Trump", or "someone else"?
Me: Donald Trump
Interviewer: Lots of things come up that keep people from attending religious services even if they want to. Thinking about your life these days, do you ever attend religious services? Please respond with "yes" or "no".
Me: **yes**

Figure 7: An interview-style context used in Study 3. The context is in plaintext; underline text shows demographic variables dynamically inserted into the interview template; one possible sampled completion is shown in bold.

(talk about politics, vote, and vote choice) information. The conditioning text included the mock interview with questions and responses for eleven of the twelve items, leaving the twelfth question for GPT-3 to answer. Each backstory was based on actual responses given by one human ANES respondent.[2] In study 2, the goal was to measure the probability of a single token; here the goal is to measure a wide variety of multi-token responses, which complicates the analysis of their raw probabilities. Instead, we allow the GPT-3 API to sample completions, which we then analyze.

Like Study 2, we use a templating system that maps ANES demographic variables to template fragments, which are then concatenated to construct the conditioning context. Because this was a virtual interview, we used phrasing that exactly matched the ANES interview verbiage whenever possible. We mapped ANES variables to short text fragments, which were then interpolated into template fragments. For this study, we used the following ANES variables (in the following order): gender (V161342), race (V161310x), age (V161267), education (V161270) church attendance (V161244), patriotism (V162125x), whether the subject discusses politics (V162174), level of interest in politics (V162256), 7-point self-reported ideology (V161126), 7-point self-reported partisanship (V161158x), whether the subject voted in 2016 (V162031x) and for whom (V162062x). Table 10 provides the full text of all ANES question wording and GPT-3 template text.

Recall that the goal of this study was to predict one factor (which we call the *target factor*, such as "Race") given specific values of all of the other factors. The template fragment for the target factor was always placed at the end of the context (and naturally did not include the corresponding ANES variable). Given a context, we asked GPT-3 to sample 5 tokens, using a temperature of 0.7. GPT-3's generated text was then lightly processed (lower-cased, stripped of leading and trailing whitespace), and compared to the limited set of ANES responses for the target factor using exact string matches. So, for example, if the target question was "Race", then the string produced by GPT-3 would be compared to "white", "black", "asian" and "hispanic", and coded as 1, 2, 3 or 5, respectively. If the GPT-3 response did not match any of the allowable responses, it was coded as missing data. This generates a dataset that is structurally equivalent to the original ANES dataset.

The combination of 12 variables and 4270 respondents resulted in the generation of more than 50,000 unique conditioning texts for GPT-3, each designed to elicit one "silicon" respondent's answer to one question. In our analysis, we keep only the 1782 observations that are complete in both ANES and GPT-3 responses. This prevents variation in the set of cases from introducing additional statistical noise to the comparison.

[2]Missing responses in the ANES data resulted in the removal of the question from the conditioning text.

Table 10: Comparison of ANES question wording and GPT-3 Template

Order	Variable	ANES VarID	ANES Question Wording	GPT-3 Template Text
1	Gender	V161342	What is your gender?	What is your gender? Are you "male" or "female"?
2	Race / Ethnicity	V161310x	I am going to read you a list of five race categories. Please choose one or more races that you consider yourself to be: - white, - black or African-American, - American Indian or Alaska Native, - Asian, or - Native Hawaiian or other Pacific Islander? + Are you Spanish, Hispanic, or Latino?	I am going to read you a list of four race categories. What race do you consider yourself to be? "White", "Black", "Asian", or "Hispanic"?
3	Age	V161247	(Derived variable - no question text)	What is your age in years?
4	Education	V161270	What is the highest level of school you have completed or the highest degee you have received?	What is the highest level of school you have completed, or the highest degree you have received? Is it "high school", "some college", "a four-year college degree", or "an advanced degree"?

continued on next page...

...continued from last page.

Order	Variable	ANES VarID	ANES Question Wording	GPT-3 Template Text
5	Attends Church	V161244	Lots of things come up that keep people from attending religious services even if they want to. Thinking about your life these days, do you ever attend religious services, apart from occasional weddings, baptisms or funerals?	Lots of things come up that keep people from attending religious services even if they want to. Thinking about your life these days, do you ever attend religious services? Please respond with "yes" or "no".
6	Patriotism	V162125x	When you see the American flag flying does it make you feel good, bad, or neither good nor bad? + Does it make you feel [extremely good, moderately good, or a little good / a little good, moderately good, or extremely good]? / Does it make you feel [extremely bad, moderately bad, or a little bad / a little bad, moderately bad, or extremely bad]?	When you see the American flag flying, how does it make you feel? Does it make you feel "extremely good", "moderately good", "a little good", "neither good nor bad", "a little bad", "moderately bad", or "extremely bad"?
7	Discusses Politics	V162174	Do you ever discuss politics with your family or friends?	Do you ever discuss politics with your family and friends? Please respond with "Yes" or "No".

continued on next page...

...continued from last page.

Order	Variable	ANES VarID	ANES Question Wording	GPT-3 Template Text
8	Political Interest	V162256	How interested would you say you are in politics? Are you [very interested, somewhat interested, not very interested, or not at all interested / not at all interest, not very interested, somewhat interested, or very interested]?	How interested would you say you are in politics? Are you "very interested", "somewhat interested", "not very interested", or "not at all interested"?
9	Voted in 2016	V162031x	In talking to people about elections, we often find that a lot of people were not able to vote because they weren't registered, they were sick, or they just didn't have time. Which of the following statements best describes you: One, I did not vote (in the election this November), Two, I thought about voting this time, but didn't, Three, I usually vote, but didn't this time, or Four, I am sure I voted? + (Derived from other Pre and Post Election Questions)	Did you vote in the 2016 general election? Please answer with "yes" or "no".

continued on next page...

...continued from last page.

Order	Variable	ANES VarID	ANES Question Wording	GPT-3 Template Text
10	2016 Vote Choice	V162062x	Who did you vote for? [Hillary Clinton, Donald Trump / Donald Trump, Hillary Clinton], Gary Johnson, Jill Stein, or someone else? + (Derived from other Pre and Post Election Questions)	Which presidential candidate did you vote for in the 2016 presidential election, "Hillary Clinton", "Donald Trump", or "someone else"? *Note: Only displayed if respondent voted.*
11	Ideology	V161126	Where would you place yourself on this scale, or haven't you thought much about this? (Scale card shown or online response options)	When asked about your political ideology, would you say you are "extremely liberal", "liberal", "slightly liberal", "moderate", "slightly conservative", "conservative", or "extremely conservative"?
12	Party ID	V161158x	Generally speaking, do you usually think of yourself as [a Democrat, a Republican / a Republican, a Democrat], an independent, or what? + Would you call yourself a strong [Democrat / Republican] or a not very strong [Democrat / Republican]? OR Do you think of yourself as closer to the Republican Party or to the Democratic Party?	Which would you say best describes your partisan identification. Would you say you are a "strong democrat", "not very strong democrat", "independent, but closer to the Democratic party", "independent", "independent, but closer to the Republican party", "not very strong Republican", or "strong Republican"?

4.2 Data analysis

The complete set of synthetic responses are appended together to create a single dataset that includes the ANES values for all eleven variables and the silicon responses for all eleven variables.

As an important methodological note, we do not calculate the direct individual-level correspondence between the ANES value for a given respondent and the GPT-3 value based on the same backstory information (such as a percent correctly predicted). GPT-3 draws tokens from a distribution of words, and we also assume distributions in outcomes in the general population. Therefore, even if GPT-3 and ANES values are drawn from the same distribution, we cannot expect them to match in any given case. The important demonstration for our point is not whether GPT-3 can correctly predict an individual, but rather whether it can produce a distribution of generated responses that is comparable to the distribution in the human data.

We use the CramerV function of the R package 'DescTools' to calculate the Cramer's V between every possible combination of the 12 variables. We use Cramer's V as it is amenable to calculation using categorical data, and, like Pearson's Chi-squared on which it is based, relies on marginal values to account for variations in base rates. Cramer's V has a range of 0 to 1. Higher values of Cramer's V indicate that knowing the value of one variable gives you more information about the likely value of the second variable.

Tables 11 - 13 report the Cramer's V values for Figure 6 in the main text of the paper.

4.2.1 Missing Data

In all presented analysis, the data are restricted to just the cases that are complete - meaning there are valid response values for all ANES *and* GPT-3 variables. Of the 4270 cases in the 2016 ANES data file, 1781 are complete cases used in the analysis.

Table 14 displays the percent of cases with missing data for each variable. GPT-3 was able to produce a valid and compliant answer in more than 85 percent of the cases for all question items, and three of the items had compliance rates above 99 percent. The percent of missing data produced varies substantially, for both humans and GPT-3. ANES data had an average of 9.3 percent missing responses for the 11 items, while GPT-3 averaged only 2.7 percent missing responses.

4.2.2 Descriptive Statistics

Table 15 presents the descriptive statistics for the variables used in Study 3, separately by data source (ANES humans or GPT-3 silicon sample).

ANES "Input"	"Output" Variable	ANES Cramer's V	GPT-3 Cramer's V	Difference
age	church.goer	0.2	0.18	0.02
age	discuss.politics	0.21	0.21	0
age	race	0.20	0.2	0
age	education	0.24	0.20	0.04
age	gender	0.18	0.2	-0.02
age	ideology	0.23	0.2	0.03
age	patriotism	0.20	0.21	-0.01
age	pid7	0.22	0.20	0.02
age	political.interest	0.22	0.21	0.01
age	vote.2016	0.24	0.22	0.02
church.goer	age	0.2	0.2	0
church.goer	discuss.politics	0.01	0.14	-0.13
church.goer	race	0.09	0.04	0.05
church.goer	education	0.06	0.01	0.05
church.goer	gender	0.04	0.02	0.02
church.goer	ideology	0.28	0.12	0.16
church.goer	patriotism	0.2	0.05	0.15
church.goer	pid7	0.22	0.19	0.03
church.goer	political.interest	0.04	0.08	-0.04
church.goer	vote.2016	0.19	0.24	-0.05
discuss.politics	age	0.21	0.22	-0.01
discuss.politics	church.goer	0.01	0.18	-0.17
discuss.politics	race	0.13	0.02	0.11
discuss.politics	education	0.2	0.11	0.09
discuss.politics	gender	0	0.08	-0.08
discuss.politics	ideology	0.16	0.06	0.1
discuss.politics	patriotism	0.02	0.1	-0.08
discuss.politics	pid7	0.16	0.11	0.05
discuss.politics	political.interest	0.4	0.28	0.12
discuss.politics	vote.2016	0.11	0.2	-0.09
race	age	0.2	0.2	0
race	church.goer	0.09	0.06	0.03
race	discuss.politics	0.13	0.05	0.08
race	education	0.1	0.07	0.03
race	gender	0.07	0.07	0.00
race	ideology	0.11	0.1	0.01
race	patriotism	0.17	0.07	0.10
race	pid7	0.18	0.1	0.08
race	political.interest	0.06	0.11	-0.05
race	vote.2016	0.17	0.11	0.06

Table 11: Cramer's V values

ANES "Input"	"Output" Variable	ANES Cramer's V	GPT-3 Cramer's V	Difference
education	age	0.24	0.23	0.01
education	church.goer	0.06	0.09	-0.03
education	discuss.politics	0.2	0.07	0.13
education	race	0.1	0.04	0.06
education	gender	0.04	0.04	0
education	ideology	0.12	0.09	0.03
education	patriotism	0.09	0.05	0.04
education	pid7	0.11	0.08	0.03
education	political.interest	0.11	0.07	0.04
education	vote.2016	0.14	0.08	0.06
gender	age	0.18	0.21	-0.03
gender	church.goer	0.04	0.04	0
gender	discuss.politics	0	0.01	-0.01
gender	race	0.07	0.03	0.04
gender	education	0.04	0.07	-0.03
gender	ideology	0.13	0.14	-0.01
gender	patriotism	0.06	0.06	0
gender	pid7	0.16	0.1	0.06
gender	political.interest	0.12	0.04	0.08
gender	vote.2016	0.09	0.11	-0.02
ideology	age	0.23	0.2	0.03
ideology	church.goer	0.28	0.07	0.21
ideology	discuss.politics	0.16	0.08	0.08
ideology	race	0.11	0.09	0.02
ideology	education	0.12	0.1	0.02
ideology	gender	0.13	0.12	0.01
ideology	patriotism	0.22	0.14	0.08
ideology	pid7	0.37	0.32	0.05
ideology	political.interest	0.15	0.14	0.01
ideology	vote.2016	0.4	0.28	0.12
patriotism	age	0.20	0.18	0.02
patriotism	church.goer	0.2	0.05	0.15
patriotism	discuss.politics	0.02	0.08	-0.06
patriotism	race	0.17	0.09	0.08
patriotism	education	0.09	0.07	0.02
patriotism	gender	0.06	0.09	-0.03
patriotism	ideology	0.22	0.14	0.08
patriotism	pid7	0.19	0.15	0.04
patriotism	political.interest	0.08	0.17	-0.09
patriotism	vote.2016	0.25	0.15	0.1

Table 12: Cramer's V values

ANES "Input"	"Output" Variable	ANES Cramer's V	GPT-3 Cramer's V	Difference
pid7	age	0.22	0.21	0.01
pid7	church.goer	0.22	0.11	0.11
pid7	discuss.politics	0.16	0.13	0.03
pid7	race	0.18	0.07	0.11
pid7	education	0.11	0.12	-0.01
pid7	gender	0.16	0.12	0.04
pid7	ideology	0.37	0.32	0.05
pid7	patriotism	0.19	0.15	0.04
pid7	political.interest	0.12	0.16	-0.04
pid7	vote.2016	0.47	0.37	0.11
political.interest	age	0.22	0.2	0.02
political.interest	church.goer	0.04	0.11	-0.07
political.interest	discuss.politics	0.4	0.16	0.24
political.interest	race	0.06	0.04	0.02
political.interest	education	0.11	0.07	0.05
political.interest	gender	0.12	0.11	0.01
political.interest	ideology	0.15	0.1	0.05
political.interest	patriotism	0.08	0.16	-0.08
political.interest	pid7	0.12	0.12	0
political.interest	vote.2016	0.12	0.12	0
vote.2016	age	0.24	0.23	0.01
vote.2016	church.goer	0.19	0.19	0
vote.2016	discuss.politics	0.11	0.23	-0.12
vote.2016	race	0.17	0.06	0.11
vote.2016	education	0.14	0.13	0.01
vote.2016	gender	0.09	0.19	-0.1
vote.2016	ideology	0.4	0.33	0.07
vote.2016	patriotism	0.25	0.16	0.09
vote.2016	pid7	0.47	0.37	0.10
vote.2016	political.interest	0.12	0.2	-0.08

Table 13: Cramer's V values

Variable	ANES	GPT-3
Age	2.8	4.7
Attends Church	0.4	0
Discusses Politics	14.6	0
Race	5.6	0.1
Education	1	14.3
Gender	1.2	0
Ideology	22.7	4.2
Patriotism	14.6	0.7
Party ID	0.5	3.6
Political Interest	14.8	1.1
2016 Vote and Choice	23.8	.52

Table 14: Percent of Observations Coded as Missing

4.3 Alternative Specifications

4.3.1 Completely Synthetic Data

The data generation process results in one vector of synthetic data based on the ANES inputs for the other eleven items. When these synthetic vectors are combined, the result is a complete dataset of synthetic data. In the main text of the paper, the Cramer's V is calculated using the ANES "input" variable and the GPT-3 output. This provides the most direct comparison between the ANES and GPT-3 results, as they are both based on the same values for one half of the Cramer's V calculations.

However, we can also estimate the Cramer's V between the various synthetic vectors, removing ANES data from the GPT-3 relationship calculation entirely. Figure 8 shows the same data for the "Human" responses, but replaces the Cramer's V between ANES and GPT-3 that forms the "GPT-3" response in the main text with a Cramer's V calculation based entirely on synthetic data. Even though the use of synthetic data in both parts introduces additional noise in the estimation, the pattern of Cramer's V comparisons is highly similar to that seen when ANES inputs are used.

4.3.2 GPT-3 Temperature Variation

Additionally, when generating the GPT-3 results, the temperature setting can be varied. Temperature controls the amount of random variation allowed in the text sampling process used by GPT-3. In the main text, we use the industry standard temperature of .7. However, to demonstrate that the results are robust to multiple samples using different settings, we also provide a replication using temperature settings of 0.001 and 1.0. A

Variable	Source	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
Age	ANES	1,781	50.1	17.5	18	35	64	90
Age	GPT-3	1,781	35.5	12.6	0	27	41	99
Attends Church	ANES	1,781	0.6	0.5	0	0	1	1
Attends Church	GPT-3	1,781	0.6	0.5	0	0	1	1
Talks Politics	ANES	1,781	0.9	0.3	0	1	1	1
Talks Politics	GPT-3	1,781	0.9	0.3	0	1	1	1
Ideology	ANES	1,781	4.1	1.6	1	3	5	7
Ideology	GPT-3	1,781	4.0	1.7	1	3	5	7
Patriotism	ANES	1,781	2.0	1.3	1	1	2	7
Patriotism	GPT-3	1,781	1.5	0.9	1	1	2	7
Party ID	ANES	1,781	3.9	2.2	1	2	6	7
Party ID	GPT-3	1,781	4.4	2.2	1	2	6	7
Political Interest	ANES	1,781	2.0	0.8	1	1	2	4
Political Interest	GPT-3	1,781	1.7	0.9	1	1	2	4
White	ANES	1,781	0.8	0.4	0	1	1	1
White	GPT-3	1,781	1.0	0.2	0	1	1	1
Hispanic	ANES	1,781	0.1	0.3	0	0	0	1
Hispanic	GPT-3	1,781	0.001	0.03	0	0	0	1
Asian	ANES	1,781	0.03	0.2	0	0	0	1
Asian	GPT-3	1,781	0.002	0.04	0	0	0	1
Black	ANES	1,781	0.1	0.3	0	0	0	1
Black	GPT-3	1,781	0.02	0.2	0	0	0	1
Some College	ANES	1,781	0.3	0.5	0	0	1	1
Some College	GPT-3	1,781	0.6	0.5	0	0	1	1
Graduate Degree	ANES	1,781	0.2	0.4	0	0	0	1
Graduate Degree	GPT-3	1,781	0.002	0.04	0	0	0	1
Bachelor's Degree	ANES	1,781	0.3	0.4	0	0	1	1
Bachelor's Degree	GPT-3	1,781	0.3	0.4	0	0	1	1
High School	ANES	1,781	0.2	0.4	0	0	0	1
High School	GPT-3	1,781	0.1	0.3	0	0	0	1
Male	ANES	1,781	0.5	0.5	0	0	1	1
Male	GPT-3	1,781	0.8	0.4	0	1	1	1
Voted in 2016	ANES	1,781	0.9	0.3	0	1	1	1
Voted in 2016	GPT-3	1,781	0.8	0.4	0	1	1	1
Trump Voter	ANES	1,553	0.4	0.5	0	0	1	1
Trump Voter	GPT-3	1,483	0.2	0.4	0	0	0	1
Clinton Voter	ANES	1,553	0.5	0.5	0	0	1	1
Clinton Voter	GPT-3	1,483	0.2	0.4	0	0	0	1
Other Voter	ANES	1,553	0.1	0.3	0	0	0	1
Other Voter	GPT-3	1,483	0.5	0.5	0	0	1	1

Table 15: Study 3 Descriptive Statistics for ANES and GPT-3 Data

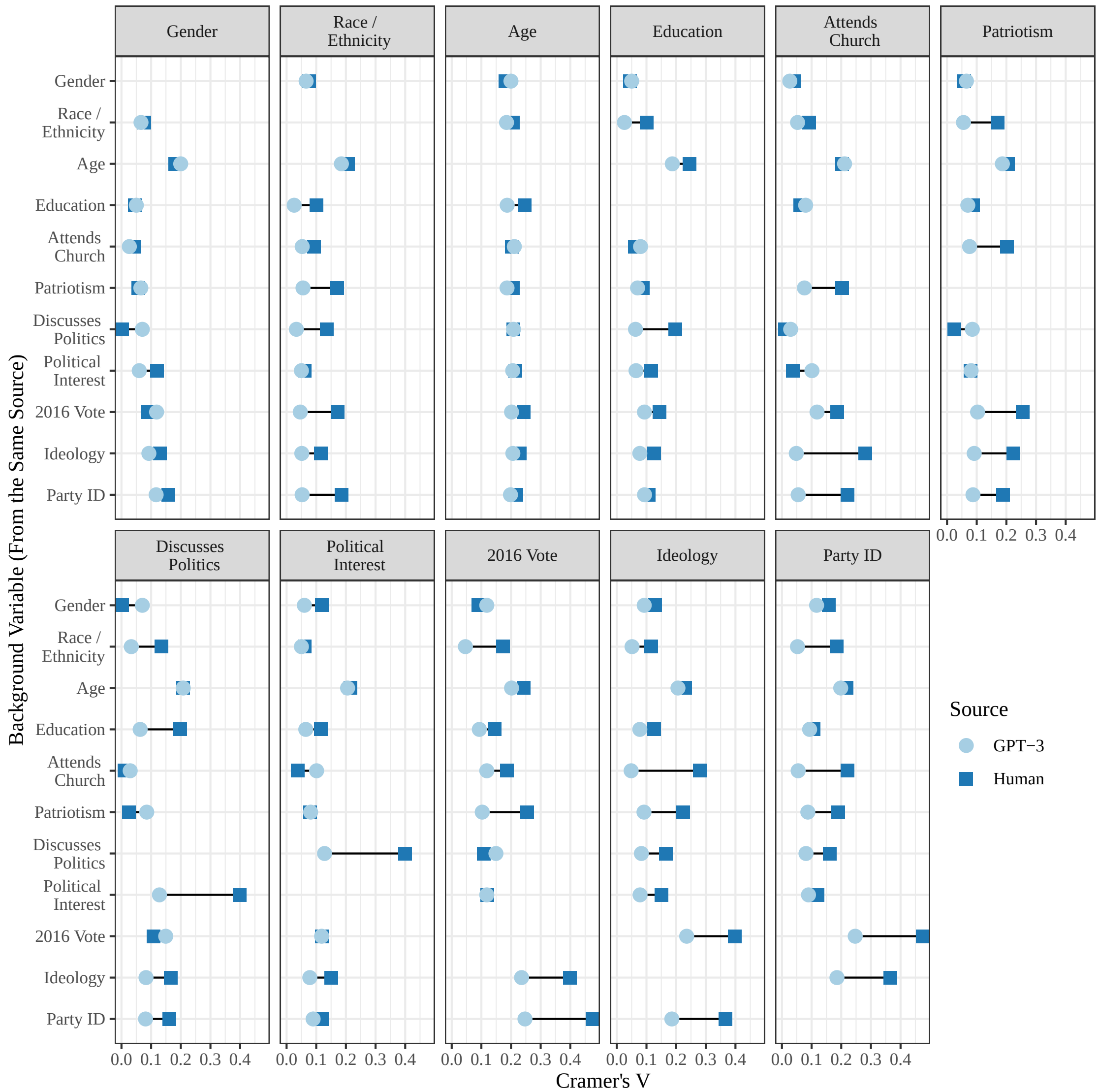

Figure 8: Cramer's V Correlations in ANES vs. GPT-3 Data, using Entirely Synthetic GPT-3 Calculations

setting of 0.001 means that in virtually all completions the algorithm will provide the response with the highest probability (meaning for a .49 to .51 split, all completions would return the token associated with the .51 probability). A setting of 1.0 means that the probability of selecting any particular token is equivalent to the probability distribution (i.e. there is no adjustment).

Table 16 shows summary statistics of the difference in Cramer's V between human and GPT-3 produced responses based on varying temperature settings. In other words,

Summary Statistic	Temp: 0.001	Temp: 0.7	Temp: 1.0
Mean	0.048	0.026	.029
Minimum	0.72	-0.17	-0.12
Maximum	0.15	0.24	0.27
Standard Deviation	0.14	0.067	0.08
N	2419	1781	1022

Table 16: Average Error in Cramer's V Based on Varying Temperatures

the value for Cramer's V produced with human data is subtracted from the value for Cramer's V produced with GPT-3 data. These results mirror the main text, and are based on ANES "inputs" and GPT-3 "outputs." Summary statistics are then calculated based on the differences. We see that, of the three options, a temperature setting of .7 produces the lowest difference between Human and GPT-3 relationships. There are minimal differences between a temperature of .7 and a temperature of 1.0. We ran each temperature query once, and did not select the presented models for best fit from a range of probabilistic runs. The results provide some evidence that the relationship patterns uncovered by GPT-3 are robust to variations in model specification.

A temperature of 0.001 produces more error and also systematically overstates the relationship between the human input and GPT-3 output variables. One caveat: at a temperature of 0.001, GPT-3 identified all respondents as white. Without variation in this variable, we were unable to calculate Cramer's V, so the GPT-3 race / ethnicity predictions are excluded from the calculations for a temperature of 0.001.

All Cramer's V calculations use the set of cases that have non-missing data for all human and GPT-3 produced variables. The different temperature settings produce a different number of valid completed cases. Lower temperature is more deterministic, and so minimizes the number of invalid tokens used as text completions. Higher temperatures sample from a range of tokens that includes more invalid responses. Therefore, mid-range temperatures appear to produce the desirable balance between validity and completeness.

5 Cost Analysis

The GPT-3 and Jurassic models are available only through a paid API. In the interests of full transparency, we here report the costs for Studies 1, 2 and 3. We only report costs for the final runs, but note that additional runs were performed as part of the experimental rhythm.

Study 1 required 1,1471 model queries (one for each human subject). The backstories were relatively small, at an average of 66 tokens. For each query, we generated a maximum of 128 tokens from the model. At the standard rate of \$0.06 / 1,000 tokens, this experiment cost a total of \$29.

Study 2 consisted of 3 experiments, one for ANES 2012, 2016 and 2020. We ran one model query per participant (5,914 in 20112, 4,270 in 2016 and 5,442 in 2020), for a total of 15,626 queries. Backstories were a bit longer than in Study 1, at an average of 80 tokens, but we only needed to generate one token per query, incurring a total cost of \$75.

Study 3 was more expensive. Because of the extended interview format, each prompt required an average of 458 tokens. For each query, we generated a maximum of 5 tokens. We performed one query for each ANES participant, for a total of 4,270 queries, resulting in a total per experiment of \$119. However, recall that Study 3 involved 12 different experiments (systematically predicting one backstory element from the others), and so the total cost of Study 3 was \$1,428.

We briefly note that using the Jurassic model (not available at the time of writing) would have reduced costs for all experiments. Also, these costs represent just the runs for the studies listed in the paper. We spent additional funds prior to these studies experimenting with different contexts.

6 GPT-3 Related Code Listings

In this section, we include code listings relevant for generating data using GPT-3. The output of these listings can then be analyzed using the replication code in the online replication repository.

6.1 Study 1: Generating words describing the outparty

This code reads data from the original Pigeonholing Partisan's dataset, constructs a backstory for each participant, and queries GPT-3 for a response.

```
import time
import numpy as np
import openai
from tqdm import tqdm
import pickle
import re
import sys

#
# ======================================================================
#

openai.api_key = "PUT YOUR API KEY HERE"

def do_query( prompt, max_tokens=2 ):
    response = openai.Completion.create(
        engine="davinci",
        prompt=prompt,
        temperature=0.7,
```

```
        max_tokens=max_tokens,
        top_p=1,
        logprobs=100,
    )
    time.sleep(1.0) # to avoid rate limiters
    return response

#
# ========================================================================
#

def uniqvals( users, field ):
    vals = [ users[id][field] for id in users.keys() ]
    return list(set(vals))

fields_of_interest = {
    "Gender": { "Male":"male", "Female":"female", '':'' },
    "Hisp":{ "Hispanic":"Hispanic", "Not Hispanic":'', '':'' },
    "WHITE":{ "White":"white", "Non-white":"", '':'' },
    "Ideo": {
        '':'',
        'Liberal':'liberal',
        'Slightly conservative':'slightly conservative',
        'Conservative':'conservative',
        'Slightly liberal':'slightly liberal',
        "Moderate/Haven't thought about it":'moderate',
        'Extremely Liberal':'extremely liberal',
        'Extremely conservative':'extremely conservative',
        },
    "PID7": {
        '':'',
        'Ind':'am an independent',
        'Strong D':'am a strong Democrat',
        'Strong R':'am a strong Republican',
        'Lean D':'lean towards Democrats',
        'Lean R':'lean towards Rebublicans',
        'Weak D':'am a weak Democrat',
        'Weak R':'am a weak Republican',
        },

    "Inc": {
        '':'',
        'Less than $15K':'very poor',
        '$15K to $25K':'poor',
        '$25K to $50K':'poor',
        '$50K to $75K':'middle-class',
        '$75K to $100K':'middle-class',
        '$100K to $150K':'middle-class',
        '$150K to $200K':'upper-class',
        '$200K to $250K':'upper-class',
        '$250K to $500K':'upper-class',
        'Prefer not to answer':'',
        '-8':'',
    },
    }

def mapper( profile ):
    results = {}
    for k in profile.keys():
        if k in fields_of_interest:
            results[k] = fields_of_interest[k][profile[k]]
```

```python
    if profile['Age'] != '':
        age = int( profile['Age'] )
        if age >= 18 and age < 25:
            results['Age'] = 'young'
        if age >= 25 and age < 40:
            results['Age'] = 'middle-aged'
        if age >= 40 and age < 60:
            results['Age'] = 'old'
        if age >= 60 and age < 100:
            results['Age'] = 'very old'
    else:
        results['Age'] = ''

    return results


lines = open( "../human_data/Pigeonholing Partisans.csv", "r" ).readlines()
dmap = {}
header_vals = lines[0].strip().split(",")
for line in lines[1:]:
    parts = line.strip().split(",")
    if len(parts) != len(header_vals):
        print("Error on line: " + line)
        continue
    dmap[ parts[0] ] = {}
    for hind,h in enumerate(header_vals):
        dmap[ parts[0] ][ header_vals[hind] ] = parts[hind]

party_list = ["Republican", "Democratic"]

results = {}
ids = dmap.keys()
for id in tqdm( ids ):
    results[id] = {}

    user_profile = mapper( dmap[id] )

    for party in party_list:

        print( f"------------------------------------------ {id} {party}" )

        prompt = ""

        if user_profile['Ideo'] != '':
            prompt += "Ideologically, I describe myself as " + user_profile['Ideo'] + ". "

        if user_profile['PID7'] != '':
            prompt += "Politically, I " + user_profile['PID7'] + ". "

        if user_profile['WHITE'] == 'White':
            prompt += "Racially, I am white. "

        if user_profile['Hisp'] == 'Hispanic':
            prompt += "Racially, I am Hispanic. "

        if user_profile['Gender'] != '':
            prompt += "I am " + user_profile['Gender'] + ". "

        if user_profile['Inc'] != '':
            prompt += "Financially, I am " + user_profile['Inc'] + ". "

        if user_profile['Age'] != '':
```

```
            prompt += "In terms of my age, I am " + user_profile['Age'] + ". "

        prompt += "When I am asked to write down four words that typically describe people"
        prompt += "who support the " + party + " Party, I respond with: 1."

        print( prompt )

        full_response = do_query( prompt, max_tokens=128 )
        text = full_response['choices'][0]['text']
        text = "1." + text  # since "1." is part of the prompt

        print(text)

        results[id][party] = text

pickle.dump( results, open( f"../gpt3_data/gpt3_synthetic_words.pkl", "wb" ) )
```

6.2 Study 1: Analyzing words generated by GPT-3

This code uses simple regular expressions to analyze the response from GPT-3.

```
import numpy as np
import pickle
import re

def extract_phrase( text, ind ):
    match = re.search( str(ind) + '.[ \t]+([A-Za-z /()-]+)([0-9]|$|\n|,|\.)', text )
    if match is None:
        return ""
    else:
        return match.group(1)

def extract_four( text ):
    # text = text.lower()
    text = re.sub( ' ', ' ', text )  # replace alternative space
    text = re.sub('[^A-Za-z0-9,. /\n\'-]', '', text)
    results = []
    for tmp in range(4):
        single_phrase = extract_phrase( text, tmp+1 )
        results.append( single_phrase )
    return results

party_list = ["Republican", "Democratic"]

results = pickle.load( open( f"../gpt3_data/gpt3_synthetic_words.pkl","rb") )

fout = open( "./gpt3_results_full.csv", "w" )

for id in results.keys():
    print( "**********************************************************************************")
    print( id )
    print( "**********************************************************************************")

    d_text = results[id][party_list[1]]
    d_results = extract_four( d_text )
    print( "==============================================")
    print( d_text )
    print( "----------------------------------------------")
    print( d_results )
```

```
    r_text = results[id][party_list[0]]
    r_results = extract_four( r_text )
    print( "==============================================")
    print( r_text )
    print( "----------------------------------------------")
    print( r_results )


    print()

    print( f"{id}" +
           "," + ",".join( [ '"'+x+'"' for x in d_results ] ) +
           "," + ",".join( [ '"'+x+'"' for x in r_results ] ),
           file=fout )
```

6.3 Study 2: Vote Prediction - Common Analytics

This code (named "common.py") contains common analytic routines:

```
import openai
import numpy as np
import time

openai.api_key = "PUT YOUR KEY HERE"

def lc( t ):
    return t.lower()

def uc( t ):
    return t.upper()

def mc( t ):
    tmp = t.lower()
    return tmp[0].upper() + t[1:]

def gen_variants( toks ):
    results = []

    variants = [ lc, uc, mc ]

    for t in toks:
        for v in variants:
            results.append( " " + v(t) )

    return results

def logsumexp( log_probs ):
    log_probs = log_probs - np.max(log_probs)
    log_probs = np.exp(log_probs)
    log_probs = log_probs / np.sum( log_probs )
    return log_probs

def extract_probs( lp ):
    lp_keys = list( lp.keys() )
    ps = [ lp[k] for k in lp_keys ]
    ps = logsumexp( np.asarray(ps) )
    vals = [ (lp_keys[ind], ps[ind]) for ind in range(len(lp_keys)) ]

    vals = sorted( vals, key=lambda x: x[1], reverse=True )
```

```
    result = {}
    for v in vals:
        result[ v[0] ] = v[1]

    return result

def do_query( prompt, max_tokens=2, engine="davinci" ):
    response = openai.Completion.create(
        engine=engine,
        prompt=prompt,
        temperature=0.7,
        max_tokens=max_tokens,
        top_p=1,
        logprobs=100,
    )

    token_responses = response['choices'][0]['logprobs']['top_logprobs']

    results = []
    for ind in range(len(token_responses)):
        results.append( extract_probs( token_responses[ind] ) )

    return results, response

def collapse_r( response, toks ):
    total_prob = 0.0
    for t in toks:
        if t in response:
            total_prob += response[t]
    return total_prob

def print_response( template_val, tok_sets, response ):
    #print( f"{template_val}" )

    print( tok_sets )

    tr = []
    for tok_set_key in tok_sets.keys():
        toks = tok_sets[tok_set_key]
        full_prob = collapse_r( response[0], toks )
        tr.append( full_prob )
        #print( f";{tok_set_key};{full_prob}", end="" )
        #print( "\t{:.2f}".format(full_prob), end="" )
    print("\t\t",end="")
    tr = np.asarray( tr )
    tr = tr / np.sum(tr)
    for ind, tok_set_key in enumerate( tok_sets.keys() ):
        print( f"\t{tok_set_key}\t{tr[ind]}", end="" )
        #print( "\t{:.2f}".format(tr[ind]), end="" )
    print("")

def parse_response( template_val, tok_sets, response ):
    tr = []
    for tok_set_key in tok_sets.keys():
        toks = tok_sets[tok_set_key]
        full_prob = collapse_r( response[0], toks )
        tr.append( full_prob )
    tr = np.asarray( tr )
    tr = tr / np.sum(tr)

    results = {}
    for ind, tok_set_key in enumerate( tok_sets.keys() ):
```

```
        results[ tok_set_key ] = tr[ind]
    return results


def run_prompts( prompts, tok_sets, engine="davinci" ):
    results = []
    for prompt in prompts:
        #print("---------------------------------------------------")
        #print( prompt )
        response, full_response = do_query( prompt, max_tokens = 2, engine=engine )
        #print( response )
        #print_response( prompt, tok_sets, response )
        simp_results = parse_response( prompt, tok_sets, response )
        #print( simp_results )
        time.sleep( 0.1 )
        results.append( (simp_results, response, full_response) )
    return results

def run_experiment( template, template_vals, tok_sets ):
    prompts = []
    for template_val in template_vals:
        grounded_prompt = template.replace( "XXX", template_val )
        prompts.append( grounded_prompt )
    return run_prompts( prompts, tok_sets )
```

6.4 Study 2: Vote Prediction - Common Templates

This code (named "anes_common.py") contains common data structures for templatizing the ANES data:

```
fips_state_map = {
    1:"Alabama",
    2:"Alaska",
    4:"Arizona",
    5:"Arkansas",
    6:"California",
    8:"Colorado",
    9:"Connecticut",
    10:"Delaware",
    # 11:"DC",
    12:"Florida",
    13:"Georgia",
    15:"Hawaii",
    16:"Idaho",
    17:"Illinois",
    18:"Indiana",
    19:"Iowa",
    20:"Kansas",
    21:"Kentucky",
    22:"Louisiana",
    23:"Maine",
    24:"Maryland",
    25:"Massachusetts",
    26:"Michigan",
    27:"Minnesota",
    28:"Mississippi",
    29:"Missouri",
    30:"Montana",
    31:"Nebraska",
    32:"Nevada",
```

```
    33:"New Hampshire",
    34:"New Jersey",
    35:"New Mexico",
    36:"New York",
    37:"North Carolina",
    38:"North Dakota",
    39:"Ohio",
    40:"Oklahoma",
    41:"Oregon",
    42:"Pennsylvania",
    44:"Rhode Island",
    45:"South Carolina",
    46:"South Dakota",
    47:"Tennessee",
    48:"Texas",
    49:"Utah",
    50:"Vermont",
    51:"Virginia",
    53:"Washington",
    54:"West Virginia",
    55:"Wisconsin",
    56:"Wyoming",
    }
```

6.5 Study 2: Vote Prediction - ANES 2012 template

This code (named "anes2012.py") contains templates for templatizing the ANES 2012 dataset:

```
from common import *
from anes_common import *

SEP=','
OUTPUT_FN = "./full_results_2012.pkl"
OUTPUT_CSV = "./full_results_2012.csv"
ID_COL = "caseid"
VOTE_COL = "presvote2012_x"
K1 = "romney"
K2 = "obama"

tok_sets = {
    "romney": gen_variants( [ "romney", "mitt", "republican", "conservative" ] ),
    "obama": gen_variants( [ "obama", "barack", "democrat", "democratic", "liberal"] ),
    }

query = "In the 2012 presidential election, I voted for"

ANES_FN = './2012 ANES.csv'

fields_of_interest = {
    # race
    'dem_raceeth_x': {
        "template":"Racially, I am XXX.",
        "valmap":{ 1:'white', 2:'black', 3:'asian', 4:'native American', 5:'hispanic' },
        },

    # discuss_politics
    'discuss_disc': {
        "template":"XXX",
        "valmap":{1:'I like to discuss politics with my family and friends.',
```

```
            2:'I never discuss politics with my family or friends.'},
    },

# ideology
'libcpre_self': {
    "template":"Ideologically, I am XXX.",
    "valmap":{
    1:"extremely liberal",
    2:"liberal",
    3:"slightly liberal",
    4:"moderate",
    5:"slightly conservative",
    6:"conservative",
    7:"extremely conservative"},
    },

# party
'pid_x': {
    "template":"Politically, I am XXX.",
    "valmap":{
    1:"a strong democrat",
    2:"a weak Democrat",
    3:"an independent who leans Democratic",
    4:"an independent",
    5:"an independent who leans Republican",
    6:"a weak Republican",
    7:"a strong Republican"},
    },

# church_goer
'relig_church': {
    "template":"I XXX.",
    "valmap":{ 1:"attend church", 2:"do not attend church"},
    },

# age
'dem_age_r_x': {
    "template":"I am XXX years old.",
    "valmap":{},
    },

# gender
'gender_respondent_x': {
    "template":"I am a XXX.",
    "valmap":{ 1:"man", 2:"woman"},
    },

# political_interest
'paprofile_interestpolit': {
    "template":"I am XXX interested in politics.",
    "valmap":{1:"very", 2:"somewhat", 3:"not very", 4:"not at all"},
    },

# patriotism
'patriot_flag': {
    "template":"It makes me feel XXX to see the American flag.",
    "valmap":{
    1:"extremely good",
    2:"moderately good",
    3:"a little good",
    4:"neither good nor bad",
    5:"a little bad",
```

```
        6:"moderately bad",
        7:"extremely bad"},
        },

    # this is sample address, which may be different than registration address...?
    'sample_stfips': {
        "template":"I am from XXX.",
        "valmap":fips_state_map,
        },

    }
```

6.6 Study 2: Vote Prediction - ANES 2016 template

This code (named "anes2016.py") contains templatizing information for the ANES 2016 dataset:

```
from common import *
from anes_common import *

OUTPUT_FN = "./full_results_2016.pkl"
OUTPUT_CSV = "./full_results_2016.csv"
SEP='|'
ID_COL = "V160001_orig"
VOTE_COL = "V162062x"
K1 = "trump"
K2 = "clinton"

tok_sets = {
    "trump": gen_variants( [ "trump", "donald", "republican", "conservative" ] ),  # the republican, mr trump
    "clinton": gen_variants( [ "clinton", "hillary", "rodham", "senator", "democrat", "democratic", "liberal"] ),
    }

query = "In the 2016 presidential election, I voted for"

ANES_FN = '../anes2016/anes_timeseries_2016_rawdata.txt'

fields_of_interest = {
    # race V161310x 1= white 2= black 3 = asian 5 = hispanic
    'V161310x': {
        "template":"Racially, I am XXX.",
        "valmap":{ 1:'white', 2:'black', 3:'asian', 4:'native American', 5:'hispanic' },
        },

    # discuss_politics V162174 1=yes discuss politics, 2=never discuss politics
    'V162174': {
        "template":"XXX",
        "valmap":{
        1:'I like to discuss politics with my family and friends.',
        2:'I never discuss politics with my family or friends.'},
        },

    # ideology V161126 1-7 = extremely liberal, ..., extremely conservative
    'V161126': {
        "template":"Ideologically, I am XXX.",
        "valmap":{
        1:"extremely liberal",
        2:"liberal",
        3:"slightly liberal",
        4:"moderate",
```

```
    5:"slightly conservative",
    6:"conservative",
    7:"extremely conservative"},
    },

# party V161158x
'V161158x': {
    "template":"Politically, I am XXX.",
    "valmap":{
    1:"a strong democrat",
    2:"a weak Democrat",
    3:"an independent who leans Democratic",
    4:"an independent",
    5:"an independent who leans Republican",
    6:"a weak Republican",
    7:"a strong Republican"},
    },

# church_goer V161244
'V161244': {
    "template":"I XXX.",
    "valmap":{ 1:"attend church", 2:"do not attend church"},
    },

# age V161267
'V161267': {
    "template":"I am XXX years old.",
    "valmap":{},
    },

# gender  V161342 1=male  2=female
'V161342': {
    "template":"I am a XXX.",
    "valmap":{ 1:"man", 2:"woman"},
    },

# political_interest = if_else(V162256 > 0, V162256, NA_real_),
'V162256': {
    "template":"I am XXX interested in politics.",
    "valmap":{1:"very", 2:"somewhat", 3:"not very", 4:"not at all"},
    },

# patriotism = if_else(V162125x > 0, V162125x, NA_real_))
'V162125x': {
    "template":"It makes me feel XXX to see the American flag.",
    "valmap":{
    1:"extremely good",
    2:"moderately good",
    3:"a little good",
    4:"neither good nor bad",
    5:"a little bad",
    6:"moderately bad",
    7:"extremely bad"},
    },

# this is sample address, which may be different than registration address...?
'V161010d': {
    "template":"I am from XXX.",
    "valmap":fips_state_map,
    },

}
```

6.7 Study 2: Vote Prediction - ANES 2020 Template

This code (named "anes2020.py") contains templates for the ANES 2020 data:

```
from common import *
from anes_common import *

SEP=','
OUTPUT_FN = "./full_results_2020.pkl"
OUTPUT_CSV = "./full_results_2020.csv"
ID_COL = "V200001"
VOTE_COL = "V202110x" # 1-Biden, 2-Trump, 3-Jorgensen, 4-Hawkins
K1 = "trump"
K2 = "biden"

tok_sets = {
    "trump": gen_variants( [ "donald", "trump", "republican", "conservative" ] ),
    "biden": gen_variants( [ "joe", "joseph", "biden", "democrat", "democratic", "liberal"] ),
    }

query = "In the 2020 presidential election, Donald Trump is the Republican candidate, "
query += "and Joe Biden is the Democratic candidate, and I voted for"

ANES_FN = './2020 ANES.csv'
fields_of_interest = {
    # race  1= white 2= black 3 = asian 5 = hispanic
    'V201549x': {
        "template":"Racially, I am XXX.",
        "valmap":{ 1:'white', 2:'black', 3:'asian', 4:'native American', 5:'hispanic' },
        },

    # discuss_politics 1=yes discuss politics, 2=never discuss politics
    'V202022': {
        "template":"XXX",
        "valmap":{
        1:'I like to discuss politics with my family and friends.',
        2:'I never discuss politics with my family or friends.'},
        },

    # ideology 1-7 = extremely liberal, ..., extremely conservative
    'V201200': {
        "template":"Ideologically, I am XXX.",
        "valmap":{
        1:"extremely liberal",
        2:"liberal",
        3:"slightly liberal",
        4:"moderate",
        5:"slightly conservative",
        6:"conservative",
        7:"extremely conservative"},
        },

    # party
    'V201231x': {
        "template":"Politically, I am XXX.",
        "valmap":{
        1:"a strong democrat",
        2:"a weak Democrat",
        3:"an independent who leans Democratic",
        4:"an independent",
        5:"an independent who leans Republican",
        6:"a weak Republican",
```

```
        7:"a strong Republican"},
        },

    # church_goer
    'V201452': {
        "template":"I XXX.",
        "valmap":{ 1:"attend church", 2:"do not attend church"},
        },

    # age
    'V201507x': {
        "template":"I am XXX years old.",
        "valmap":{},
        },

    # gender 1=male  2=female
    'V201600': {
        "template":"I am a XXX.",
        "valmap":{ 1:"man", 2:"woman"},
        },

    # political_interest = if_else(V162256 > 0, V162256, NA_real_),
    'V202406': {
        "template":"I am XXX interested in politics.",
        "valmap":{1:"very", 2:"somewhat", 3:"not very", 4:"not at all"},
        },

    # this is sample address, which may be different than registration address...?
    'V201007d': {
        "template":"I am from XXX.",
        "valmap":fips_state_map,
        },

    }
```

6.8 Study 2: Vote Prediction - Main predictor

This code generates vote predictions from ANES data, as well as cost estimates:

```
import sys
import pandas as pd
import pickle
from tqdm import tqdm

# for cost analysis
from transformers import GPT2Tokenizer

if sys.argv[1] == '2012':
    from anes2012 import *
if sys.argv[1] == '2016':
    from anes2016 import *
if sys.argv[1] == '2020':
    from anes2020 import *

from common import *

foi_keys = fields_of_interest.keys()

#
# ==========================================================================================
```

```python
# ==========================================================================================
#

def cost_approximation(prompt, engine="davinci", tokenizer=None):
    possible_engines = ["davinci", "curie", "babbage", "ada"]
    assert engine in possible_engines, f"{engine} is not a valid engine"

    if tokenizer==None:
        tokenizer = GPT2Tokenizer.from_pretrained("gpt2")

    num_tokens = len(tokenizer(prompt)['input_ids'])

    if engine == "davinci":
        cost = (num_tokens / 1000) * 0.0600
    elif engine == "curie":
        cost = (num_tokens / 1000) * 0.0060
    elif engine == "babbage":
        cost = (num_tokens / 1000) * 0.0012
    else:
        cost = (num_tokens / 1000) * 0.0008

    return cost, num_tokens

def gen_backstory( pid, df ):
    person = df.iloc[pid]

    backstory = ""

    for k in foi_keys:
        anes_val = person[k]
        elem_template = fields_of_interest[k]['template']
        elem_map = fields_of_interest[k]['valmap']

        if len(elem_map) == 0:
            backstory += " " + elem_template.replace( 'XXX', str(anes_val) )

        elif anes_val in elem_map:
            backstory += " " + elem_template.replace( 'XXX', elem_map[anes_val] )

    if backstory[0] == ' ':
        backstory = backstory[1:]

    return backstory

#
# ==========================================================================================
# ==========================================================================================
#

anesdf = pd.read_csv( ANES_FN, sep=SEP, encoding='latin-1' )

costs = []
numtoks = []
tokenizer = GPT2Tokenizer.from_pretrained("gpt2")

full_results = []
for pid in tqdm( range(len(anesdf)) ):

    if "V200003" in anesdf.iloc[pid] and anesdf.iloc[pid]["V200003"]==2:
        print( f"SKIPPING {pid}..." )
        # we want to exclude cases marked as 2 on this variable;
        # those are the panel respondents (interviewed in 2016 and 2020)
```

```
        continue

    anes_id = anesdf.iloc[pid][ID_COL]

    prompt = gen_backstory( pid, anesdf )
    prompt += " " + query

    #print("--------------------------------------------------")
    #print( prompt )

    cost, numtok = cost_approximation( prompt, engine="davinci", tokenizer=tokenizer )
    costs.append( cost )
    numtoks.append( numtok )

    results = run_prompts( [prompt], tok_sets, engine="davinci" )
    #print(results[0][0])
    full_results.append( (anes_id, prompt, results) )

print( "Total cost: ", np.sum(np.array(costs)) )
print( "Averge numtok: ", np.mean(np.array(numtoks)) )

pickle.dump( full_results, open(OUTPUT_FN,"wb") )
```

6.9 Study 3: Second Order Correlations - ANES 2020 Template

To replicate Study 3, an entire set of ANES data must be generated for each target demographic variable of interest; we used the following shell script:

```
#!/bin/bash

python ./openai_test_interview.py gender
python ./openai_test_interview.py race
python ./openai_test_interview.py age
python ./openai_test_interview.py education
python ./openai_test_interview.py patriotism
python ./openai_test_interview.py discuss_politics
python ./openai_test_interview.py political_interest
python ./openai_test_interview.py ideology
python ./openai_test_interview.py pid7
python ./openai_test_interview.py votechoice_2016
python ./openai_test_interview.py voted_2016
python ./openai_test_interview.py church_goer
```

which then calls the following main script. This generates each "interview" and then queries GPT-3:

```
import sys
import numpy as np
import openai
import time
import scipy.stats as stats
import pandas as pd
import pickle
from tqdm import tqdm

# for cost analysis
from transformers import GPT2Tokenizer
```

```python
from common import *  # reuse from study2 code

openai.api_key = "PUT YOUR KEY HERE"

age_map = {}
for ind in range(100):
    age_map[ind]=str(ind)

questions = {

    'V161342': {
        'desc':'gender',
        'vals': {1:'male',2:'female'},
        'question': 'Interviewer: What is your gender? Are you "male" or "female"?',
        },

    'V161310x': {
        'desc':'race',
        'vals': {1:'white',2:'black',3:'asian',5:'hispanic'},
        'question': 'Interviewer: I am going to read you a list of four race categories. What race do you ' + \
                    'consider yourself to be? "White", "Black", "Asian", or "Hispanic"?',
        },

    'V161267': {
        'desc':'age',
        'vals': age_map,
        'question': 'Interviewer: What is your age in years?',
        },

    'V161270': {
        'desc':'education',
        'vals': {
            1:'high school',
            2:'high school',
            3:'high school',
            4:'high school',
            5:'high school',
            6:'high school',
            7:'high school',
            8:'high school',
            9:'high school',
            10:'some college',
            11:'some college',
            12:'some college',
            13:'a four-year college degree',
            14:'an advanced degree',
            15:'an advanced degree',
            16:'an advanced degree',
        },
        'question': 'Interviewer: What is the highest level of school you have completed, or the highest ' + \
                    'degree you have received? Is it "high school", "some college", "a four-year college ' + \
                    'degree", or "an advanced degree"?',
        },

    'V161244': {
        'desc':'church_goer',
        'vals': {1:'yes',2:'no'},
        'question': 'Interviewer: Lots of things come up that keep people from attending religious ' + \
                    'services even if they want to. Thinking about your life these days, do you ever' + \
                    'attend religious services? Please respond with "yes" or "no".',
        },
```

```
'V162125x': {
    'desc':'patriotism',
    'vals': {1:"extremely good", 2:"moderately good", 3:"a little good",
             4:"neither good nor bad", 5:"a little bad", 6:"moderately bad", 7:"extremely bad"},
    'question': 'Interviewer:  When you see the American flag flying, how does it make you feel? Does ' + \
                'it make you feel "extremely good", "moderately good", "a little good", "neither good ' + \
                'nor bad", "a little bad", "moderately bad", or "extremely bad"?'
    },

'V162174': {
    'desc':'discuss_politics',
    'vals': {1:'yes',2:'no'},
    'question': 'Interviewer: Do you ever discuss politics with your family and friends? Please respond ' + \
                'with "Yes" or "No".'
    },

'V162256': {
    'desc':'political_interest',
    'vals': {1:"very interested", 2:"somewhat interested", 3:"not very interested", 4:"not at all interested"},
    'question': 'Interviewer: How interested would you say you are in politics? Are you "very interested", ' + \
                '"somewhat interested", "not very interested", or "not at all interested"?',
    },

'V161126': {
    'desc':'ideology',
    'vals': {1:"extremely liberal", 2:"liberal", 3:"slightly liberal",
             4:"moderate", 5:"slightly conservative", 6:"conservative", 7:"extremely conservative"
    },
    'question': 'Interviewer: When asked about your political ideology, would you say you are "extremely '+ \
                'liberal", "liberal", "slightly liberal", "moderate", "slightly conservative", ' + \
                '"conservative", or "extremely conservative"?',
    },

'V161155': {
    'desc':'pid3',
    'vals': {1:'Democrat',2:'Republican',3:'Independent'},
    'question': 'Interviewer: Generally speaking, do you usually think of yourself as a "Democrat", a ' + \
                '"Republican", or an "Independent"?',
    },

'V161158x': {
    'desc':'pid7',
    'vals': {1:"strong democrat", 2:"not very strong democrat",
             3:"independent, but closer to the Democratic party", 4:"independent",
             5:"independent, but closer to the Republican party", 6:"not very strong Republican",
             7:"strong Republican"},
    'question': 'Interviewer: Which would you say best describes your partisan identification. ' + \
                'Would you say you are a "strong democrat", "not very strong democrat", ' + \
                '"independent, but closer to the Democratic party", "independent", "independent, ' + \
                'but closer to the Republican party", "not very strong Republican", or "strong Republican"?',
    },

'V162031x': {
    'desc':'voted_2016',
    'vals': {0:'no',1:'yes'},
    'question': 'Interviewer: Did you vote in the 2016 general election? Please answer with "yes" or "no".'
    },

'V162062x': {
    'desc':'votechoice_2016',
    'vals': {1:"Hillary Clinton", 2:"Donald Trump", 42:"someone else"},
    'question': 'Interviewer: Which presidential candidate did you vote for in the 2016 presidential ' + \
```

```
        'election, "Hillary Clinton", "Donald Trump", or "someone else"?',
        },

    }

#
# ==========================================================================================
#

def render_question( s, q, last_q = False ):
    txt = ''

    if q == 'V161155':

        if s['V161155'] == 2:
            txt += 'Interviewer: Thinking about your identification with the Republican party, would you say ' + \
                   'it is "strong" or a "not very strong"?\n'
            if last_q:
                txt += f"Me:"
                return txt

            if s['V161156'] == 1:
                txt += f"Me: strong\n\n"
            elif s['V161156'] == 2:
                txt += f"Me: not very strong\n\n"

        if s['V161155'] == 1:
            txt += 'Interviewer: Thinking about your identification with the Democratic party, would you say ' + \
                   'it is "strong" or a "not very strong"?\n'
            if last_q:
                txt += f"Me:"
                return txt

            if s['V161156'] == 1:
                txt += f"Me: strong\n\n"
            elif s['V161156'] == 2:
                txt += f"Me: not very strong\n\n"

        if s['V161155'] == 'Independent':
            txt += 'Interviewer: Do you think of yourself as "closer to the Republican Party", "closer to the ' + \
                   'Democratic party", or "closer to neither party"?\n
'
            if last_q:
                txt += f"Me:"
                return txt

            if s['V161157'] == 1:
                txt += "Me: closer to the Republican Party\n\n"
            elif s['V161157'] == 2:
                txt += "Me: closer to neither party\n\n"
            elif s['V161157'] == 3:
                txt += "Me: closer to the Democratic party\n\n"


    # XXX note, we don't check to make sure the ANES answer is "valid"
    if last_q:
        txt += questions[q]['question'] + "\n"
        txt += f"Me:"
        return txt

    if s[q] in questions[q]['vals']:
        txt += questions[q]['question'] + "\n"
```

```
        txt += f"Me: {questions[q]['vals'][ s[q] ]}\n\n"
        return txt
    else:
        return txt

def find_q( questions, hrq ):
    for q in questions.keys():
        if questions[q]['desc'] == hrq:
            return q
    error("not found!")

def build_interview( s, human_readable_omit=None ):
    txt = ''

    human_readable_question_order = [ 'gender', 'race', 'age', 'education', 'church_goer',
              'patriotism', 'discuss_politics', 'political_interest', 'ideology', 'pid7',
              'voted_2016', 'votechoice_2016' ]

    omit = None
    if human_readable_omit:
        omit = find_q( questions, human_readable_omit )

    for hrq in human_readable_question_order:
        q = find_q( questions, hrq )
        if q == omit:
            continue
        if hrq=='votechoice_2016' and s['V162031x'] == 0:
            continue

        txt += render_question( s, q, last_q=False )

    if human_readable_omit:
        txt += render_question( s, omit, last_q=True )

    return txt

#
# ==========================================================================================
# ==========================================================================================
#

def cost_approximation(prompt, engine="davinci", tokenizer=None):
    possible_engines = ["davinci", "curie", "babbage", "ada"]
    assert engine in possible_engines, f"{engine} is not a valud engine"

    if tokenizer==None:
        tokenizer = GPT2Tokenizer.from_pretrained("gpt2")

    num_tokens = len(tokenizer(prompt)['input_ids'])

    if engine == "davinci":
        cost = (num_tokens / 1000) * 0.0600
    elif engine == "curie":
        cost = (num_tokens / 1000) * 0.0060
    elif engine == "babbage":
        cost = (num_tokens / 1000) * 0.0012
    else:
        cost = (num_tokens / 1000) * 0.0008

    return cost, num_tokens

#
```

```
# ===========================================================================================
# ===========================================================================================
#

def strcompare( s1, s2 ):
    s1 = s1.lower().strip()
    s2 = s2.lower().strip()
    return s1.startswith(s2) or s2.startswith(s1)

print( "===================================================" )
print( "RUNNING WITH " + sys.argv[1] )
print( "===================================================" )


df = pd.read_csv( '../anes2016/anes_timeseries_2016_rawdata.txt', sep='|')

final_results = []

#hr_omit = 'votechoice_2016'
#hr_omit = 'church_goer'
hr_omit = sys.argv[1]

omit = find_q( questions, hr_omit )

costs = []
numtoks = []
tokenizer = GPT2Tokenizer.from_pretrained("gpt2")

for ind, row in tqdm( df.iterrows() ):
    #print( "===================================================\n" )

    id = row['V160001_orig']

    prompt = build_interview( row,  human_readable_omit=hr_omit )

    cost, numtok = cost_approximation( prompt, engine="davinci", tokenizer=tokenizer )
    costs.append( cost )
    numtoks.append( numtok )

    #print(prompt )
    full_results = do_query( prompt, max_tokens = 5, temperature=0.001 )

    samp_response = full_results[1]['choices'][0]['text']

    #clean_r = samp_response.replace('\n',' ')
    #print( f"{questions['V161158x']['vals'][row['V161158x']]} -> {clean_r}", end='' )

    coded_response = -1
    for valnum in questions[omit]['vals'].keys():
        if strcompare( questions[omit]['vals'][valnum], samp_response ):
            #print( f" -> {questions[omit]['vals'][valnum]} -> {valnum}", end='' )
            coded_response = valnum
    #print('')

    final_results.append( {
      "id":id,
      "prompt":prompt,
      "sampled_response":samp_response,
      "coded_response":coded_response,
      "full_results":full_results} )

print( "Total cost: ", np.sum(np.array(costs)) )
```

```python
print( "Averge numtok: ", np.mean(np.array(numtoks)) )

pickle.dump( final_results, open("./heatmap_backstory_" + hr_omit + "_full_results.pkl","wb") )

# ------------------

fout = open( "./heatmap_" + hr_omit + "_results.csv", "w" )

keys = ['V160001_orig'] + list(questions.keys())

print( ",".join(keys) + ",gpt3_coded_response", file=fout )

for ind, row in tqdm( df.iterrows() ):
    print( ",".join( [str(row[k]) for k in keys] ) + f",{final_results[ind]['coded_response']}", file=fout )

fout.close()
```