CENTER FOR CAMPAIGN INNOVATION

RESEARCH REPORT

From Open-Ends to Insights: Leveraging AI in Survey Response Analysis

How campaigners can use AI to analyze open-ended survey responses faster and more affordably.

Discussions about the use of artificial intelligence (AI) focus primarily on the imagined harms of widespread synthetic images and videos. But this fixation ignores the many areas where AI can be used responsibly to improve the ways voters and campaigns interact with each other.

AI excels at data analysis and natural language processing. Campaigns can tap into these capabilities to improve the effectiveness of survey research and polling analysis. What previously would have taken many hours at great expense – often prohibitive for campaigns – is now achievable within just a few minutes with the help of AI.

In a typical survey, researchers would ask a voter "What is the most urgent issue facing the country today?" and provide a handful of choices like "the economy," "climate change," "immigration," and "abortion".

By providing options to respondents in this fashion – making it easy to analyze – pollsters are "leading the witness" and limit the insights that can be gleaned from voters.

For example, we don't know if a respondent who answered "abortion" is pro-life or pro-choice or whether the "international conflict" they're concerned about is the war in Ukraine or the war in Gaza.

The alternative approach of asking the question as an open-end allows a pollster to detect these nuances, but reviewing and categorizing each response is time consuming and expensive. AI tools, however, are adept at identifying and categorizing text based on context – even with missing keywords, poor grammar, or misspellings.

In a recent survey we conducted in a battleground congressional district, we decided to ask voters about their most important issue via an open-ended question rather than multiple choice. We then used AI to analyze and categorize the responses.

We received 432 responses to the question, "In your own words, what is the most important issue you think the US Congress should be working on right now?" Using LLMs we created code books and applied them to each response. We also compared various models to see how they performed.

Models Tested

ChatGPT

Grok

Gemini

Claude

Part I: Creating A Prompt

Our first step was to ask each model to generate a prompt that would help us create the code books and perform the analysis.

We prompted ChatGPT o4-mini-high, Grok 3, Gemini 2.5 Pro, and Claude Opus 4 each with the following instruction:

```
I need you to write a prompt. I
want to use an LLM to code open-
ended responses to a survey.

I will provide a CSV with
responses and the model needs to
identify any topic that occurs at
least 3 times in the data set.
Each response may be coded with
multiple topics if needed. Our
objective is to get as many
responses coded as possible.

I need the output as a CSV with
each row being a response and each
code being its own column with a 1
in any column code related to the
response.
```

The requirement to produce a CSV with each code as a column is based on initial testing where the LLMs would provide unreliable counts. This output allows for easy double checking of the model's arithmetic.

Each model generated a unique prompt to complete the task while preserving the key requirements.

Claude Opus 4

The Claude Opus 4-generated prompt was the longest of the prompts we explored. It included examples of the types of codes that would be "Clear and descriptive." It also stated that codes should be "specific enough to be meaningful but broad enough to capture variations of the same theme."

Further down in the prompt, it included examples of inputs and outputs with references to a more commercial customer survey, like "product quality," customer service," and "affordability."

Its instructions included these guidelines:

- Be inclusive in your coding - if a response relates to a topic even tangentially, code it
- Prioritize coding as many responses as possible over being overly strict
- If you identify a topic that appears only 1-2 times, do not create a code for it
- Topic names should use lowercase with underscores between words
- Ensure every response is checked against every identified topic
- If a response doesn't fit any topic that meets the 3+ occurrence threshold, it should have all 0s

ChatGPT o4-mini-high

o4-mini-high began with the instruction "you are an expert qualitative data coder" then gave it a series of steps numbered one through six. Its specific instruction about coding included:

- Topics should be concise (2-4 words) and formatted consistently in Title Case (e.g. "Healthcare Costs").
- Exclude filler words or one-off adjectives ("and," "against," "agenda," etc.).
- Group synonyms or near-duplicates under a single topic label.

The prompt also included the step "Aim to code as many responses as possible with at least one topic."

Grok 3

Grok 3's prompt provided a structure outlined by Task, Input, Instructions, and Objective rather than a numerical list. It provided less detailed instructions about what the topics should or should not be like the ChatGPT prompt. It did include the text "To code as many responses as possible with the identified topics" as the Objective.

Gemini 2.5 Pro

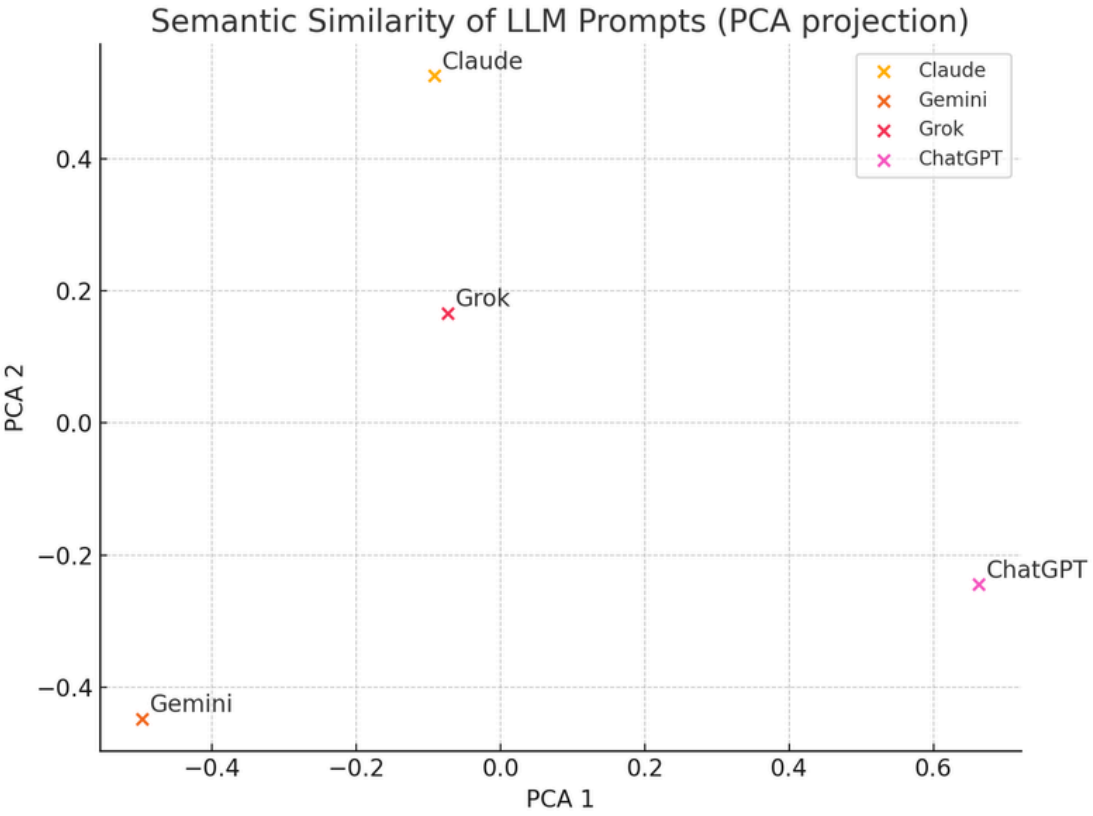
Like the ChatGPT o4-mini-high prompt, Gemini 2.5 Pro began with assigning a role: “you are an expert data analyst,” described the task, then outlined instructions numbered one through four. Like the Claude Opus 4 prompt, it similarly included example inputs and outputs with references to commercial categories.

Analysis

To analyze the similarities and differences among the four prompts, we asked ChatGPT to apply a quantitative text analysis using Term Frequency–Inverse Document Frequency (TF-IDF) and cosine similarity.

Each prompt was converted into a vector representation that captured the relative importance of words within and across the texts. Then, it calculated pairwise cosine similarity scores between these vectors to assess how semantically similar the prompts were.

The results showed that the Claude and Grok prompts were the most similar (similarity score of 0.59), while ChatGPT’s prompt was the most distinct, showing lower similarity with the others, especially Gemini (0.31).



Part II: Testing The Prompts

Next, we proceeded to prompt each model with each prompt to compare the results. ChatGPT and Grok provided the type of responses we expected as you will see below.

Using Claude's prompt in Claude resulted in the chatbot producing a Python script that it instructed us to use to complete the analysis rather than producing it in the chat. When we tried the ChatGPT prompt in Claude, we received a message saying "This response paused because Claude reached its max length for a message. Hit continue to nudge Claude along."

After selecting continue, we received the message again. Selecting continue a second time resulted in the message "You've reached the limit for Claude messages at this time. Please wait before trying again." Because of these issues, we did not include Claude in our comparison.

While using Gemini, we encountered an issue where the model would generate a hyperlink to a CSV file and instructions to download it, but the file did not exist. We pointed the error out to the AI, but despite assurances the issue was fixed, the file was never generated so we were unable to include Gemini in our comparison.

Categories

How many codes were identified

Prompt	ChatGPT	Grok
Claude	20	7
ChatGPT	11	21
Grok	14	12
Gemini	15	12

Coverage

What percentage of responses were coded.

Prompt	ChatGPT	Grok
Claude	65%	76%
ChatGPT	45%	56%
Grok	75%	88%
Gemini	48%	80%

ChatGPT Responses

Ten most used codes

Claude	ChatGPT	Grok	Gemini
Immigration	Immigration	Immigration	Immigration
Trump	Economy	Economy	Economy
Economy	Taxation	Legislation & Congress	Taxes
Taxes	Jobs	Trump & Impeachment	Border Control
Border Control	Impeachment	Civil Liberties	Jobs
Legislation	Democracy	Budget & Spending	Impeachment
Constitution	Healthcare	Taxes	Democracy
Federal Budget	Social Security	Healthcare	Cost of Living
Rights	Inflation	Democracy	Healthcare
Impeachment	Abortion	Jobs	Social Security

🟩	🟨	🟧	⬜
Appears in all four	*Appears in three*	*Appears in two*	*Appears in one*

We also tasked a human on our team with no prior survey coding experience to undertake the same project. After approximately 90 minutes, she identified 23 codes, applying them 536 times across 427 responses.

Human Coded

1. Border Security / Immigration
2. Economy
3. Impeach Trump
4. Budget / National Debt
5. Democracy / Constitution
6. Balance of Power / Corruption
7. Human / Civil Rights
8. Taxes
9. Big Beautiful Bill
10. Healthcare

Analysis

The models converged on how to code the data set with slight variations in terminology, splitting categories, and assessing edge cases making it extremely efficient and quick at generating a code book for a given set of verbatims.

The AI analysis also highlights the benefits of asking this question as an open-end rather than series of multiple choices because topics like climate change and abortion that are frequently included rarely appeared in our data set, indicating that the inclusion of these topics influences the responses.

Additionally, we were surprised to see how often the Big Beautiful Bill was mentioned explicitly in the verbatims indicating awareness of the legislation prior to its passage at the time the survey was conducted.

Part III: Iterating On Our Prompts

Iteration is an essential strategy for effectively using AI. We decided to compile learnings from across the four models and begin a new round of prompting that accounted for:

- Making clear in our prompt the context of the open-ended question so the models would understand that the responses are regarding politics. This may help to focus the model and improve performance.
- Combine elements like the role assignment found in ChatGPT and Gemini with detailed instructions like those found in Claude and the clear objective of comprehensive coverage as seen in Grok.
- Turn on temporary chats to avoid preconditioning the model with previous exposure to the prompts.

For the second round of prompting, we relied exclusively on ChatGPT's models based on previous experience, our existing subscription with access to the most advanced versions, and no limits on requests.

Our new initial prompt was a more detailed and refined version of the one used in Part I. It offered clearer guidances and reduced ambiguity for the model. The updated prompt specified the exact input format, including column names, and provided the full survey question for context. It also introduced stricter rules for excluding infrequent topics (mentioned only once or twice) and instructs the model to ensure each response is evaluated against all identified topics. Additionally, the new prompt emphasizes the goal of maximizing the number of responses coded and clarifies that the output should retain the original columns alongside the new binary topic indicators.

The new coding instruction prompt added greater context, flexibility, and clarity while relaxing some of the rigid formatting requirements. Whereas the earlier prompt framed the task as a set of technical instructions for a data coder, Prompt 2 is more situational and focused on analyzing a specific question. It included a real-world example of synonymous issues to guide coding decisions and emphasized judgment and coverage over format rigidity.

After providing this prompt to ChatGPT 4o, it identified 9 categories, coded 294 response, and assigned codes 380 times.

Then, we prompted it to keep going:

```
Review all of the responses that
did not have any codes assigned
and identify additional codes.
Then repeat the earlier code
assignment process for all
responses.
```

The model identified 8 additional topics, coded a total of 323 responses, and assigned codes 437 times.

We decided to keep pressing:

```
109 responses remain uncoded.
Review them and identify any
possible additional codes. You may
assign a code "Unsure" for any
responses like "I don't know".
```

```
You may assign a code "Unsure" for
any responses like "I don't know".
Then repeat the earlier code
assignment process for all
responses.
```

This resulted in the creation of four new categories, 26 additional responses coded, and 33 new codes assigned. At this point the model began suggesting codes that did not meet the threshold of appearing three or more times and we stopped the analysis. See Table XX for the responses

Quality Check

We conducted a detailed review of the final dataset to assess the quality of the AI output and found 129 instances where human analysis disagreed:

- 19 responses had extra, incorrect codes applied
- 49 responses had an incorrect code applied
- 79 responses were missing a code that should have been applied

Analysis

Pressing the AI model to identify more codes was effective at gaining a more comprehensive overview of the data, but issues remained with certain responses being difficult for AI to categorize. There are some obvious instances where a simple keyword match would identify the correct code, which may indicate some technical

Responses

Initial Round

1. Immigration / Border Security
2. Economy / Inflation
3. Donald Trump / Impeachment / Trump-related
4. Democracy / Voting Rights
5. Taxes
6. Healthcare / Medicare
7. Government Corruption / Accountability
8. Public Safety / Crime / Law Enforcement
9. Gun Control / Second Amendment

Followup

1. Immigration / Border Security
2. Economy / Inflation
3. Donald Trump / Impeachment / Trump-related
4. Democracy / Voting Rights
5. Taxes
6. Healthcare / Medicare
7. Government Corruption / Accountability
8. Peace / Foreign Policy
9. National Debt / Fiscal Responsibility
10. Homelessness / Poverty
11. Public Safety / Crime / Law Enforcement
12. Reproductive Rights / Abortion
13. Environmental Issues / Climate Change
14. Gun Control / Second Amendment
15. Partisan Gridlock / Bipartisanship
16. Presidential Power / Balance of Power
17. Education

Final Round

1. Immigration / Border Security
2. Economy / Inflation
3. Donald Trump / Impeachment / Trump-related
4. Democracy / Voting Rights
5. Taxes
6. Healthcare / Medicare
7. Government Corruption / Accountability
8. Peace / Foreign Policy
9. General Dissatisfaction / Everything
10. National Debt / Fiscal Responsibility
11. Big Bill Legislation
12. Homelessness / Poverty
13. Public Safety / Crime / Law Enforcement
14. Authoritarianism / Threats to Democracy
15. Reproductive Rights / Abortion
16. Environmental Issues / Climate Change
17. Gun Control / Second Amendment
18. Partisan Gridlock / Bipartisanship
19. Presidential Power / Balance of Power
20. Education
21. General Governance / Representation

limitations like context window saturation or prompt decay.

Indeed, asking the model to identify new codes and assign them was more effective than simply providing the list of codes to perform the task.

Discussion

Asking the issue question as an open end rather than multiple choice gave us greater insight into how voters think and speak about these topics. For example, if respondents were only given the option of immigration, we would not have been able to distinguish a voter who says “Securing the border. The number one thing is getting all these people out of the country that don't belong here. versus a voter who says “Stopping the illegal deportations” but they would both choose immigration.

We also get a better sense of the specific language and wording voters are using to talk about these issues. This can help inform our messaging, script writing, and even training AI models to write more persuasively.

As we’ve come to expect with AI models, the results weren’t without errors or issues. In our quality review we found that it assigned incorrect codes, missed correct codes, and added extra codes to responses.

Each time we executed the analysis workflow, we observed slight variations in both the identified categories and how individual responses were coded, which raises important considerations about reliability and reproducibility when using AI for qualitative analysis.

Additionally, in order to achieve this level of analysis, we increased the cost of our survey since open ended responses take longer for participants to provide than multiple choice questions. This additional time commitment from respondents typically requires higher incentives, which impacts the overall budget for the research. Despite this higher cost, the rich qualitative data obtained through open-ended questions, now made more accessible through AI-assisted coding, provides value that may justify the increased expense in many research scenarios.

Campaigners should explore ways to incorporate more opportunities for open ended feedback into their surveys now that AI can assist with coding. With the increasing sophistication of AI tools, the traditional barriers to analyzing qualitative data—such as time constraints, resource limitations, and the need for specialized expertise—are being significantly reduced. Open-ended questions allow respondents to express their thoughts in their own words, potentially revealing nuances, priorities, and framing that might not be captured through pre-determined multiple-choice options. This approach can provide campaign strategists with more authentic insights into voter concerns and perspectives, helping to shape more responsive and effective messaging strategies.

Based on our experience, we strongly recommend allowing the AI model to independently identify the codebook and conduct several initial coding passes, followed by a comprehensive manual review. This approach leverages the AI's ability to efficiently process large volumes of qualitative data while acknowledging the necessity of human oversight to refine categories, correct misclassifications, and ensure analytical rigor. By combining the computational efficiency of AI with human contextual understanding and subject matter expertise, researchers can achieve a more balanced, accurate, and nuanced analysis of open-ended survey responses while significantly reducing the time and resource investment traditionally required for qualitative coding.

Eric Wilson
Executive Director

Scan the code to view all of the prompts used in this article.

About

The Center for Campaign Innovation is a 501(c)(4) nonprofit organization.

We test cutting-edge tactics and tools in real-world campaigns, helping political professionals adopt effective, technology-driven strategies, and foster a culture of innovation in the conservative movement.