Evaluating Large Language Models for Semi-Structured Data Manipulation Tasks: A Survey Platform Case Study

Vinicius Monteiro de Lira
vmonteirodelira@surveymonkey.com
SurveyMonkey
Padua, Italy

Antonio Maiorino
amaiorino@surveymonkey.com
SurveyMonkey
Padua, Italy

Peng Jiang
pjiang@surveymonkey.com
SurveyMonkey
San Mateo, USA

Rouzbeh Torabian Esfahani
rtorabianesfahani@surveymonkey.com
SurveyMonkey
Ottawa, Canada

Abstract

Large Language Models (LLMs) are increasingly leveraged for complex data manipulation tasks involving semi-structured data formats such as JSON. In this paper, we introduce a novel classification-based evaluation framework specifically designed to systematically assess the accuracy, and impact of LLM-driven edits on semi-structured JSON data. Our framework transforms the generative nature of LLM edits into a multi-label classification problem, enabling objective, reproducible measurement of both content-based and structural modifications. We detail the architecture of our toolkit, which combines JSON diffing, change type detection, and label mapping to generate annotated ground truth datasets and reliable evaluation metrics. To showcase the practical utility and generalizability of our approach, we apply our framework in a real-world deployment at SurveyMonkey, where LLMs facilitate interactive survey editing. Through large-scale A/B testing, integration with production metrics, and comprehensive error analysis, our results provide actionable insights into the strengths and limitations of LLMs when editing semi-structured survey schema.

CCS Concepts

• **Computing methodologies** → **Natural language generation**; • **Applied computing** → **Text editing**; **Document metadata**.

Keywords

LLMs, Data Manipulation, Semi-structured data, Evaluation

ACM Reference Format:
Vinicius Monteiro de Lira, Antonio Maiorino, Peng Jiang, and Rouzbeh Torabian Esfahani. 2025. Evaluating Large Language Models for Semi-Structured Data Manipulation Tasks: A Survey Platform Case Study. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (KDD Workshop A-GenAI Eval).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

KDD Workshop A-GenAI Eval, Toronto, Canada

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

We are a worldwide leader in survey and form software, with our premier platform facilitating the collection of over 25 million responses daily, effectively serving every use case–from startups to Fortune 500 companies. In SurveyMonkey, a key objective of our service is to support customers in designing high-quality surveys.

We have introduced a new functionality called *Interactive Edits*, an AI-powered feature that enables users to refine and modify their surveys dynamically. Through text-based prompts, users can adjust the survey tone, edit question text, update answer options, and perform various modifications, streamlining the survey design process. Figure 1 illustrates how users can interact with the platform to adjust survey content through simple text instructions. Once a user submits a prompt using the 'sent' button, it is processed by a LLM, which updates the survey structure according to the user's request.

The main challenge we faced is to optimize this interactive editing functionality to ensure seamless user interaction and accurate modifications, since evaluating generative model outputs is particularly challenging because of the free-text form typical of Survey questions which is more akin to creative writing than other Structured Data Manipulation tasks [4, 6, 13]. In this setting, measuring the effectiveness of generative models is inherently complex, since responses often require nuanced judgment and lack straightforward criteria for correctness, unlike many other natural language processing tasks.

To overcome these challenges when evaluating edits performed by an LLM, we have developed an evaluation framework that measures how accurately the model incorporates user-requested modifications. By representing surveys as semi-structured JSON data, our framework tracks structural changes — insertions, deletions, updates, and rearrangements — quantifying alignment between user intent and model output. Our key contribution is this generalizable evaluation framework, which can be applied beyond survey editing to any AI-driven data manipulation task involving semi-structured documents.

2 Related Work

Large Language Models (LLMs) have been used for a variety of applications, and there have also been some efforts to leverage them to perform and evaluate common Data Manipulation tasks [7, 9]. Most of the applications in this domain have been focused

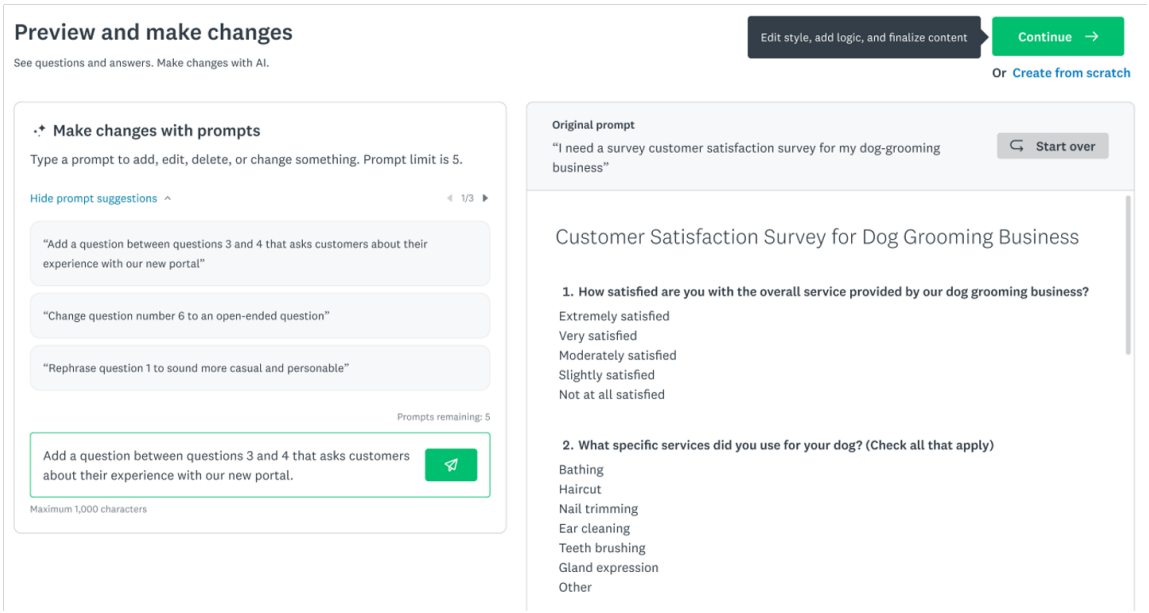

Figure 1: Interactive Edits User Interface: AI-powered editing feature allows users to modify surveys using natural language prompts.

on Structured Datasets [8, 15], especially for Data Cleaning tasks such as Data Imputation. Examples of this are [5, 11], where the authors propose methods to leverage LLMs to perform typical Data Cleaning operations with a specific focus on tabular datasets. Similarly, other works have explored applications of LLMs to perform Data Integration and Data Cleaning tasks such as Entity Matching [12] and Machine Translation [2], while there have also been some efforts focused on leveraging LLMs to clean Semi-Structured Data [8, 10] or to extract Structured information from Unstructured datasets [1] .

There have also been examples of broader efforts in the Data Manipulation domain. For example, in [14] the authors propose UniDM, a general-purpose framework for broad Data Manipulation tasks to be applied to data lakes, including Data Imputation, Transformation, Error Detection, and Entity Resolution, while the authors of [3] propose a query answering system that works over heterogeneous Data Lakes. Similarly, we also formalize a workflow combining steps produced by a Large Language Model with automated parsing and rule-based steps, focusing on a well-defined set of editing operations. However, rather than developing a tool to automate the full Data Manipulation process, we developed a framework to evaluate an LLM-based Data Manipulation module. This enabled us to iterate more effectively on analyses and improvements of our Survey Editing system.

3 Problem Definition

Semi-structured documents, such as JSON and XML, store data in a flexible format with nested attributes and varying structures. Unlike relational databases, these documents require specialized handling for data manipulation [8, 9].

In our framework, all data manipulation operations are guided by user prompts and executed by a Large Language Model (LLM). The LLM interprets each user input (prompt) to understand the user's intent and then applies changes to the survey document accordingly, while strictly preserving the underlying document schema to ensure data integrity. These editing capabilities are organized into three core types of operations:

- **Insert**: The LLM adds new elements to the survey, such as questions or answer options.
- **Update**: The LLM modifies existing content, including rewording questions, rearranging items, changing question tone, or refining answer choices.
- **Delete**: The LLM removes elements from the survey, such as questions or answer options.

We propose a framework to track and validate LLM-driven modifications in semi-structured data. Let D be a collection of documents where each $D_i \in D$ has attributes A_i. A user request, encoded as a prompt p, defines a set of tasks T (e.g., insert, update, delete) executed by an LLM function F_{LLM} that modifies records $R \subseteq D$, producing an output $Y = F_{LLM}(R, p)$.

The main challenge is verifying whether the LLM has correctly applied the requested modification to the document. This framework ensures automated tracking and validation of LLM-driven data changes, improving reliability and consistency in semi-structured data manipulation.

To address this, we focus on a case study in survey design data, specifically using our platform, as a test platform. This study examines how LLMs assist users in editing survey structures, adjusting

question formats, modifying response options, and performing various other operations while ensuring that the intended changes are accurately incorporated into the survey data.

4 Methodology: Evaluation Framework for Survey Editing

To ensure the reliability of LLM-driven survey modifications, we propose a systematic evaluation framework to assess whether user-requested edits, provided via prompts, are accurately reflected in the survey design.

4.1 Basic concept

Surveys are structured questionnaires designed to collect data from individuals, aiming to obtain insights, opinions, or feedback on a specific topic. We formally define a *survey* as follows:

Definition 4.1 (Survey). A survey consists of a set of questions intended to gather targeted information from participants. Formally, we represent a survey as a tuple $< h, l >$, where h denotes the survey title and l is the list of questions included in the survey.

Each individual *survey question* is structured as follows:

Definition 4.2 (Survey question). A survey question is represented as a tuple $< t, k, o >$, where t corresponds to the question text, k specifies its type from a predefined set K, and o denotes the collection of response options. The predefined set K includes various question formats such as open-ended questions, Net Promoter Score (NPS) questions, contact information fields, rating scales, and more.

Definition 4.3 (Response options). Response options refer to the predefined choices available for a given survey question, allowing respondents to select an appropriate answer. Formally, they are represented as a list $o = [o_1, o_2, ..., o_n]$, where each o_i corresponds to a specific selectable answer. These options vary depending on the question type and can include multiple-choice selections, Likert scales, or categorical responses.

Definition 4.4 (User prompt). A user prompt represents the user's intent when modifying an existing survey. Through text-based instructions, the user specifies desired edits to the survey structure, question formats, or response options.

4.2 Framework Architecture

The diagram in Figure 2 illustrates the evaluation framework for assessing LLM-driven survey modifications. It showcases the complete workflow, from a user prompt specifying edits to an existing survey to the mapping of applied changes, and the final assessment - using human annotations - of whether the LLM correctly incorporated the requested edits. The key idea here is to frame the generative problem as a classification problem, allowing for a more structured and actionable evaluation process.

The inference process starts with a user prompt requesting survey modifications (e.g., adding, changing, or rearranging questions). An LLM processes the JSON-formatted survey, applies the edits, and generates an updated version. This happens without human intervention and while maintaining the same output structure (schema).

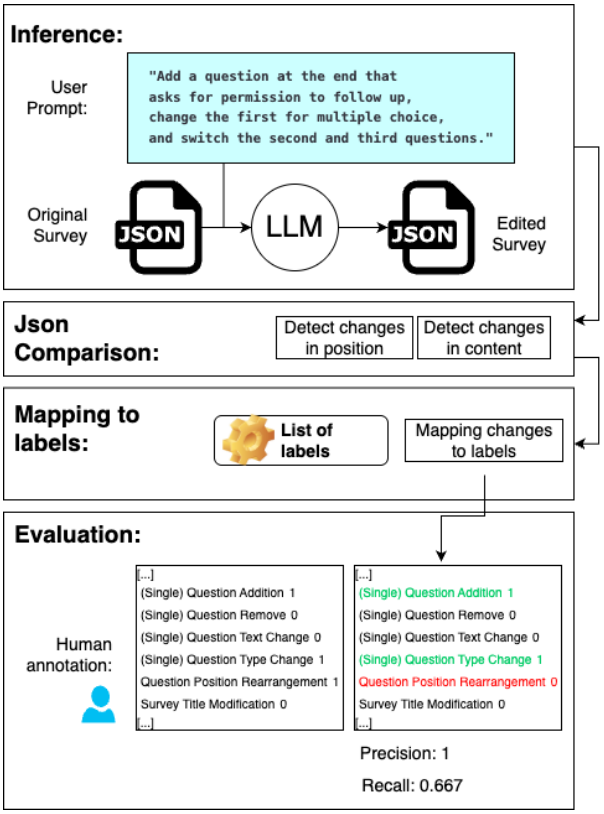

Figure 2: Framework workflow

Changes between the current and modified survey are extracted and mapped based on modifications to the JSON schema, identifying insertions, deletions, updates, and rearrangements. These detected modifications are then categorized into a **predefined set of labels** (L), which includes all relevant operations on elements (D) and attributes (A) tracked by the framework. For instance, the label "Single Question Addition" denotes the insertion of a question element, "Question Position Rearrangement" refers to repositioning an element, and "Title Modification" refers to changes to the title attribute of a question element.

The detection of changes and positions, as well as the mapping of these changes, is reflected in Algorithms 1, 2, and 3. Algorithm 1 compares the original survey data (oldSvy) with its edited counterpart (newSvy), detects differences, and maps these detected changes to a set of labels that indicate the type of modifications made. We leveraged Python libraries such as *jsondiff*[1] to compare JSON representations of the original and edited surveys. Then, Algorithm 2 checks each question in the new survey, compares its position to the edited survey, and collects any that have changed positions into a log with their old and new indices. Finally, Algorithm 3 identifies the content and position changes extracted with the previous algorithms and assigns one or more predefined labels to those changes, returning a list or vector of label flags (e.g. additions, deletions, etc.).

[1] https://jsondiff.com/

The final evaluation verifies whether each requested change was correctly applied by comparing LLM-driven modifications with human annotations. This structured validation ensures alignment with user intent and enables an objective, automated assessment of survey editing accuracy. Furthermore, it allows us to systematically evaluate and fairly compare different models and system prompts within the same framework.

Algorithm 1 Detect changes in content

```
function DETECT_CHANGES_IN_CONTENT(oldSvy, newSvy, L)
    initialize lookup_chgs dicionary
    diffs ← jsondiff(oldSvy, newSvy)
    for every difference in diffs do
        Identify updates by comparing elements and attributes
        If an element is removed, add it to the deletes
        If an element is inserted, add it to the inserts
    end for
    return lookup_chgs
end function
```

Algorithm 2 Detect changes in position

```
function DETECT_CHANGES_IN_POS(oldSvy, newSvy)
    initialize lookup_pos dictionary
    for every item in newSvy.Questions do
        if item.id appeared in oldSvy.Questions with a different
position then
            add that position change to the lookup dictionary
        end if
    end for
    return lookup_pos
end function
```

Algorithm 3 Map survey changes to labels

```
function MAP_TO_LABELS(oldSvy, newSvy, L)
    cont_chgs ← detect_changes_in_content(oldSvy, newSvy, L)
    pos_chgs ← detect_changes_in_pos(oldSvy, newSvy)
    fill mapping with appropriate labels
    return mapping values
end function
```

4.3 Ground Truth Development

To ensure accurate evaluation of LLM-generated survey edits, we constructed a ground truth dataset through an annotation process. A team of annotators reviewed survey editing requests and classified the intended modifications according to a predefined taxonomy. Each annotation task presented annotators with the following elements: (i) the **current survey design**, which included the survey title, list of questions, and respective attributes such as question type, text, and answer options; (ii) the **user prompt**, which contained text-based instructions describing the desired changes to the survey; and (iii) the **history of previous edits**, providing context for iterative modifications.

Annotators were responsible for examining the user prompt in the context of the current survey design and identifying the specific modifications requested. This was framed as a multi-label classification task, where each prompt could correspond to multiple types of edits. These modifications were categorized into distinct labels, encompassing operations such as *adding, removing*, or *modifying questions*, *rearranging question order*, *changing question types*, *editing response options*, and *modifying the survey title*. Some categories such as *Answer Options Modification* were further subdivided into cases of addition or removal. This structured labeling process ensured consistency in defining how user edits should be reflected in the modified survey, forming a reliable benchmark for evaluating the LLM's ability to execute user-driven modifications accurately.

Table 1 presents example user prompts corresponding to each label class in our survey editing taxonomy. These examples illustrate the types of modifications users may request when interacting with the AI-powered survey editing feature. It is important to note that some prompts may fit more than one label depending on their content. For example, a prompt such as "Change question 5 to a multiple-choice format and update its wording to be more concise" could be assigned both the *Question Type Change* and *Question Text Change* labels, as it involves modifying both the question type and its phrasing.

5 Experiments: Off-line Tests

To assess our approach, we conducted a comprehensive set of offline experiments leveraging real user data, annotated ground truth, and multiple large language models. The following subsections describe our experimental methodology and outline the results.

5.1 Experiment Setup

Data Collection: To evaluate the accuracy of AI-driven survey editing, we collected real user editing requests from our platform between April 2024 and August 2024. This data was gathered during our first A/B test for this feature, where a subset of users was given access to the *Interactive Edits*, allowing us to capture real-world editing interactions. A total of 10,403 user interactions were recorded, spanning 3,858 unique user sessions. Of these, we randomly sampled 2,500 unique interactions for data annotation.

Ground Truth Development: We used the process described in section 4.3 to build our gold standard dataset. Annotators were tasked with examining user prompts and identifying the specific changes requested to the survey design. For each annotation, annotators were provided with the current survey design, the history of previous prompts, and the user's latest instruction. Using this context, they classified the requested modifications according to predefined options. Each prompt had a total of 3 annotators. We used a majority voting criteria to determine the final set of labels for each sample, excluding annotations where the annotators were not confident about the intent of the users. We focused only on English prompts, excluding also improper and unsuported requests, resulting in a total of 1828 annotated prompts.

Evaluation metrics: We use precision and recall as evaluation metrics, calculated based on the ground truth provided by human

Table 1: Example Prompts for each label class

Label Class	Example Prompts
(Single) Question Addition	- Insert a question at the end that asks for permission to follow up. - Add a question around overall rating this employee's performance.
(Multiple) Question Addition	- Add two questions at the end: First, What kind of activities would you like to have during the weekend? Second, Any other comment or suggestion you would like to make? and both should have a text box for the answer.
(Single) Question Remove	- Remove the question asking about the user's age. - Drop the last question.
(Multiple) Question Remove	- Delete all the open-ended questions.
(Single) Question Text Change	- Change the question "How satisfied are you with our service?" to "How would you rate our service?" - Update the text from "What is your favorite feature?" to "Which feature do you use the most?"
(Multiple) Question Text Change	- Modify the wording of questions about product usage to be more specific. - Make the overall tone of the survey more casual and personable and fun. Stay away from using "awesome."
(Single) Question Type Change	- Change the question "How often do you use our product?" from multiple-choice to an open-ended format.
(Multiple) Question Type Change	- Change all multiple-choice questions to rating scale questions. - Modify the last two multiple-choice questions to single-choice.
Question Position Rearrangement	- I think question 4 should be first and phrase it that they are interested in receiving an early bird registration discount. - Drop the first question. - Insert a NPS question after the third question. - Remove the NPS question (if it is not the last).
Answer Options Modification	- Make bubbles along side of the multiple choice answers so students can choose an answer. - On question 3, change the online store option to lifewave website. - Edit the list of tools and materials in question number 9: to be the following: Sketchbooks, Pencil, Watercolors, Brushes, Pens, Travel Stool. (if the previous option list had the same number of elements)
Answer Options Addition	- Change testing features for question 3: remove "data security", add "device functionality", remove "emergency alerts", add "coverage performance", add "1 to 1 communication". - For question 3, add: Other (please list). - Edit the list of tools and materials in question number 9 to add: Sketchbooks, Pencil, Watercolors, Brushes, Pens, Travel Stool (if the previous option list had fewer elements).
Answer Options Remove	- Change testing features for question 3: remove "data security", add "device functionality", remove "emergency alerts", add "coverage performance", add "1 to 1 communication". - Can you remove nutrition supplements in the choices for question 2? And remove physical store on question 3. - Edit the list of tools and materials in question number 9 to: Sketchbooks, Pencil, Watercolors, Brushes, Pens, Travel Stool (if the previous option list had more elements).
Survey Title Modification	- Update the title from "Product Survey" to "Annual Product Evaluation."
Not English	- ¿Puedes agregar una pregunta sobre la satisfacción general? - Ajoutez une section concernant les suggestions des clients.
Other / Improper or Unsuported Instruction	- Add a section for future improvements. - Please fix the survey (unsupported: fix survey issues). - It should be 4 surveys, one for each company product (unsupported: multiple surveys). - Any new ideas to improve the club? (not a survey edit instruction) - Add (incomplete or unclear prompt) - daghdaghdgah (gibberish)

annotations. To generate labels from the model predictions, we follow the evaluation framework workflow outlined in the referenced section. 4.2

Models Tested: We compared several LLMs to evaluate their accuracy on user-requested modification tasks. Models tested:

- **GPT-4o**: `gpt-4o-2024-05-13`
- **GPT-4o Mini**: `gpt-4o-mini-2024-07-18`
- **GPT-3.5 Turbo** (baseline): `gpt-3.5-turbo-0125` (*this version was used during the A/B test*)

System Prompts: We tested four prompt variations to evaluate their impact on the performance of the system:

- **general** (baseline): A general-purpose prompt used during the A/B test for a user group, designed for broad data manipulation rather than survey-specific edits.
- **tailored**: A more specialized prompt tailored to the most common survey editing elements: survey title, questions, and answer options.
- **tailored_CoT**: A *tailored* version with Chain of Thought (CoT) reasoning for step-by-step decision-making.
- **tailored_history**: Similar to *tailored* but also including session history, retaining all prior user prompts within the conversation to enhance contextual understanding.

5.2 Experiment Results

Table 2 reports the performance results ordered by the weighted precision metric (W-Precision). The notation $\mathcal{B}^{\text{Model}}_{Prompt}$ represents the Interactive Edits system ($\mathcal{B}$) having as a setup the model (Model) and the prompt (*Prompt*). Based on the results, we can observe that tailored prompts had a relevant impact in the model's performance. GPT-4o with a tailored prompt had the best overall performance, with the highest precision (0.732) and recall (0.828). This shows that giving the model clear, domain-specific instructions significantly improves its ability to accurately perform survey edits. GPT-4o Mini performed similarly, making it a strong, efficient alternative. Also, adding session history lowered precision, suggesting that excessive context may hurt accuracy. General prompts underperformed, with GPT-3.5 showing the weakest results.

$\mathcal{B}^{\text{Model}}_{Prompt}$	W-Precision	W-Recall
$\mathcal{B}^{\text{gpt4o}}_{tailored}$	**0.732**	0.828
$\mathcal{B}^{\text{gpt4o}}_{tailored\_CoT}$	0.728	0.834
$\mathcal{B}^{\text{gpt4o\_mini}}_{tailored}$	0.724	0.833
$\mathcal{B}^{\text{gpt4o\_mini}}_{tailored\_CoT}$	0.717	**0.839**
$\mathcal{B}^{\text{gpt4o}}_{tailored\_history}$	0.698	0.818
$\mathcal{B}^{\text{gpt4o\_mini}}_{tailored\_history}$	0.693	0.813
$\mathcal{B}^{\text{gpt4o\_mini}}_{general}$	0.645	0.818
$\mathcal{B}^{\text{gpt35}}_{general}$	0.638	0.764
$\mathcal{B}^{\text{gpt4o}}_{general}$	0.602	0.813

Table 2: Model Performance Across Prompts

5.3 Results per label

Table 3 reports detailed performance metrics for our best approach, `GPT-4o` with the *tailored* prompt. The model demonstrates strong precision and recall across various categories (L) of survey modification, particularly excelling in structured tasks such as removing multiple questions (Precision = 0.86, Recall = 0.95) and modifying answer options (Precision = 0.63, Recall = 0.93). However, more nuanced modifications, such as changing single question text (Precision = 0.42, Recall = 0.77) exhibit lower precision, indicating challenges in accurately capturing fine-grained text changes. Notably, high recall in most categories suggests that the model is effective at applying the requested modifications but sometimes introduces unintended changes, thus reducing precision.

Category	Precision	Recall	Support
(Multiple) Question Addition	0.83	0.70	164
(Multiple) Question Remove	**0.86**	0.95	19
(Multiple) Question Text Change	0.83	0.79	150
(Multiple) Question Type Change	0.78	**1.00**	40
(Single) Question Addition	0.90	0.83	357
(Single) Question Remove	0.71	0.93	45
(Single) Question Text Change	0.42	0.77	104
(Single) Question Type Change	0.54	0.84	50
Answer Options Addition	0.73	0.83	138
Answer Options Modification	0.63	0.93	264
Answer Options Remove	0.43	0.95	57
Question Position Rearrangement	0.76	0.69	179
Survey Title Modification	0.50	**1.00**	34

Table 3: Metrics for Survey Modification Categories

6 Experiments: A/B Testing

Interactive Edits 2.0 (IE2) was introduced with substantial enhancements over the initial version, both in terms of underlying **model configuration** and **user experience**. The new release had been deployed using our best candidate model defined in our off-line experiments ($\mathcal{B}^{\text{gpt4o}}_{tailored}$). Additionally, to support users in crafting effective prompts, IE2 incorporates "kickstarters"—predefined prompt templates that address common editing actions. These improvements were designed to boost both the effectiveness and usability of the AI-powered survey editing workflow.

In this section, we examine two key production metrics for evaluating the system in real-world usage. The *acceptance rate* is based on explicit user feedback, reflecting the proportion of proposed changes that users accepted over the total number of requests. The *editing effectiveness rate* measures the model's ability to generate survey modifications, representing the percentage of user requests that resulted in actual survey editing. Requests that are unclear, vague, or lack sufficient support typically do not prompt changes in the survey. Since there is no labeled data in production, this metric assesses only whether a change was generated, not its accuracy. Together, these production metrics offer direct insights into both user satisfaction and operational system performance.

6.1 Effectiveness in Generating Survey Edits

Table 4 presents the percentage of user interactions ("predictions") that resulted in actual survey modifications ("effective editing") for two versions of the Interactive Edits (i.e. IE1 and IE2), along with the A/B Test Dates and the underlying model configuration used in each version. A proper prompt is defined as a user request that was successfully interpreted by the system and led to a change in the survey. The table highlights the significant improvement in prompt effectiveness with the transition from the baseline model to the tailored setup. For both A/B tests, the control group redirected users to the default editing page, where they could perform manual editing without the use of AI prompts.

Table 4: Effective editing: percentage of user requests that caused changes in the survey, by feature version and model.

	A/B Test	Model	Sessions	Pred.	Effect. (%)
IE1	2024-04-08 2024-08-08	$\mathcal{B}^{\text{gpt35}}_{general}$	3,858	10,403	79.6
IE2	2025-01-09 2025-03-05	$\mathcal{B}^{\text{gpt4o}}_{tailored}$	7,009	16,070	94.1

In IE1, the editing effectiveness rate was 79.6%, indicating that over 20% of user requests did not result in survey changes. This suggests either a lack of sensitivity to user intent or model limitations in processing and applying user requests. Additionally, the relatively low effectiveness rate may also be attributable to users' limited understanding of how to formulate prompts effectively for modifying the survey.

6.2 User Acceptance Study

To assess the real-world effectiveness of our system we ran another A/B test in production, exposing users from the treatment group to *Interactive Edits 2.0* with the best setup based on our evaluation ($\mathcal{B}^{\text{gpt4o}}_{tailored}$). When a user submits an edit request, the system highlights the differences between the original and modified survey, allowing the user to review and either accept or reject the changes.

This provided a direct proxy metric for assessing user satisfaction with the AI-driven modifications. The users from the control group were not exposed to the Interactive Edits feature. We define the *acceptance rate* as the percentage of changes that users accepted out of all AI-generated modifications within a session.

From the data collected between January and March 2025, we recorded a total of 16,070 interactions (spanning 7009 user sessions) with *Interactive Edits 2.0*, with an acceptance rate of 84%, indicating that users largely found the AI-generated modifications helpful and relevant to their intent.

6.3 Kickstarters Study

The following kickstarters are provided to help users formulate effective prompts for editing their surveys:

- **Add a question:** Add a question about …
- **Change the tone:** Change the tone of the survey to be …
- **Rephrase a question:** Rephrase question number …
- **Add answer options:** Add a "None of the above" answer option to question number …

These kickstarters guide users in phrasing their prompts clearly, making it easier to request specific survey edits without confusion. They reduce guesswork and help users quickly discover the range of available actions, leading to more accurate and efficient interactions. By providing structured templates, the system helps ensure prompts are explicit and easily understood by the AI.

6.3.1 Editing Effectiveness. Table 5 presents the effectiveness editing rate as a function of Kickstarter usage. The table compares two groups—those using kickstarters and those not using them—based on the number of predictions made and the corresponding effectiveness percentage. Specifically, individuals or systems utilizing kickstarters achieved a higher effectiveness rate (96.4%) across 5,969 predictions, compared to a 92.7% effectiveness observed in 10,101 predictions without kickstarters. The results indicate that kickstarters provide a slight boost to the model's editing effectiveness. However, the IE2 model already demonstrates high efficacy overall, regardless of Kickstarter usage.

Table 5: Effectiveness editing rate by kickstarter usage.

Kickstarter Usage	Predictions	Effect. (%)
Using kickstarters	5,969	96.4
Not using kickstarters	10,101	92.7

6.3.2 User Acceptance. Table 6 presents a comparison of acceptance rates for survey edits made with and without the use of kickstarter prompts. The acceptance rate measures the proportion of system-generated suggestions that users approved. Interestingly, the acceptance rate is slightly lower when kickstarters are used (0.794) compared to not using them (0.847).

Table 6: Acceptance rate by kickstarter usage.

Kickstarter	Acceptance Rate
Using kickstarters	0.794
Not using kickstarters	0.847

In turn, Table 7 breaks down acceptance rates by specific types of kickstarter prompts, showing both the acceptance rate and the number of prompts for each type. Certain prompt types perform particularly well; for instance, "Add a question" has the highest acceptance rate at 0.861, indicating that users are generally receptive to this type of automatic suggestion. Conversely, the "Change the tone" prompt yields the lowest acceptance rate at 0.671, pointing to potential problems with user expectations.

In summary, some kickstarters did not prove to be effective in improving the acceptance rate. For example, the prompt "Change the tone of the survey to be" achieved an acceptance rate of only 0.671. It may be worth considering exploring ways to improve its acceptance.

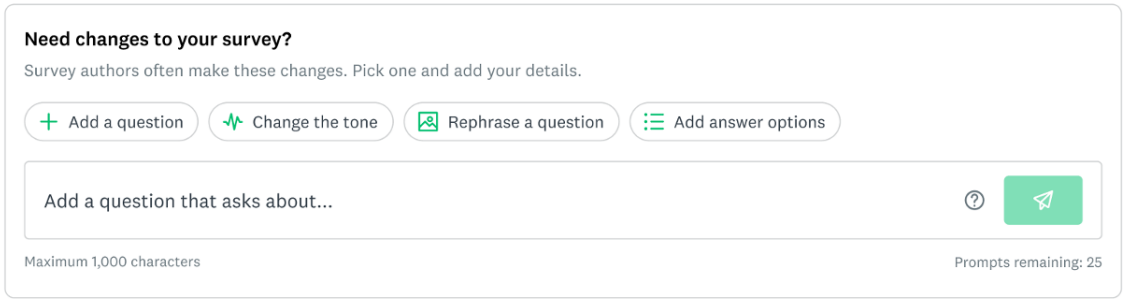

Figure 3: Interactive Edits 2.0 Kickstarters: The editing interface provides users with four kickstarter prompts—Add a question, Change the tone, Rephrase a question, and Add answer options—to streamline and guide common survey modifications.

Table 7: Acceptance rate and prompt count by kickstarter type.

Kickstarter Type	Acceptance Rate	# Prompts
Add a question	0.861	2828
Change the tone	0.671	1043
Rephrase a question	0.727	1421
Add answer options	0.795	677

7 Conclusions

In this paper, we introduced a systematic evaluation framework for assessing Large Language Models (LLMs) editing capabilities of semi-structured data, applying it to the domain of survey editing. We applied this framework to evaluate our Interactive Edits feature, which allows users to refine surveys dynamically through AI-driven modifications. Experiments demonstrate that a tailored system prompt significantly enhances model performance.

A key limitation of the framework is the difficulty of creating ground truth datasets for complex JSON structures with deeply nested elements, making annotation more challenging. As future work, we plan to integrate LLM-as-judge strategies to automate evaluation and reduce the need for human annotators. Additionally, we intend to experiment with open-source models and various fine-tuned versions to further improve performance and generalizability.

Nevertheless, by framing the generation of survey edits as a classification task, our framework introduces a systematic approach and produces results that are directly actionable. This structure not only enables consistent benchmarking across various models and prompts, but also facilitates the identification of strengths and weaknesses for each configuration when manipulating semi-structured data — more specifically, survey structure documents as the object of study presented in this work.

References

[1] Simran Arora, Brandon Yang, Sabri Eyuboglu, Avanika Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. 2023. Language models enable simple systems for generating structured views of heterogeneous data lakes. *arXiv preprint arXiv:2304.09433* (2023).

[2] Quinten Bolding, Baohao Liao, Brandon James Denis, Jun Luo, and Christof Monz. 2023. Ask language model to clean your noisy translation data. *arXiv preprint arXiv:2310.13469* (2023).

[3] Zui Chen, Zihui Gu, Lei Cao, Ju Fan, Samuel Madden, and Nan Tang. 2023. Symphony: Towards Natural Language Query Answering over Multi-modal Data Lakes.. In *CIDR*. 1–7.

[4] Vinicius Monteiro de Lira, Antonio Maiorino, and Peng Jiang. 2024. Enhancing the Reliability of LLMs-based Systems for Survey Generation through Distributional Drift Detection. In *Fourth Workshop on Knowledge-infused Learning*.

[5] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2022. Turl: Table understanding through representation learning. *ACM SIGMOD Record* 51, 1 (2022), 33–40.

[6] Michael Desmond, Zahra Ashktorab, Qian Pan, Casey Dugan, and James M Johnson. 2024. EvaluLLM: LLM assisted evaluation of generative outputs. In *Companion Proceedings of the 29th International Conference on Intelligent User Interfaces*. 30–32.

[7] Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, and Christos Faloutsos. 2024. Large Language Models (LLMs) on Tabular Data: Prediction, Generation, and Understanding–A Survey. *arXiv preprint arXiv:2402.17944* (2024).

[8] Hanbum Ko, Hongjun Yang, Sehui Han, Sungwoong Kim, Sungbin Lim, and Rodrigo Hormazabal. 2024. Filling in the gaps: Llm-based structured data generation from semi-structured scientific data. In *ICML 2024 AI for Science Workshop*.

[9] Weizheng Lu, Jing Zhang, Ju Fan, Zihao Fu, Yueguo Chen, and Xiaoyong Du. 2025. Large language model for table processing: A survey. *Frontiers of Computer Science* 19, 2 (2025), 192350.

[10] Manuel Mondal, Julien Audiffren, Ljiljana Dolamic, Gérôme Bovet, and Philippe Cudré-Mauroux. 2024. Cleaning Semi-Structured Errors in Open Data Using Large Language Models. In *2024 11th IEEE Swiss Conference on Data Science (SDS)*. IEEE, 258–261.

[11] Zan Ahmad Naeem, Mohammad Shahmeer Ahmad, Mohamed Eltabakh, Mourad Ouzzani, and Nan Tang. 2024. RetClean: Retrieval-Based Data Cleaning Using LLMs and Data Lakes. *Proceedings of the VLDB Endowment* 17, 12 (2024), 4421–4424.

[12] Avanika Narayan, Ines Chami, Laurel Orr, Simran Arora, and Christopher Ré. 2022. Can foundation models wrangle your data? *arXiv preprint arXiv:2205.09911* (2022).

[13] Arjun Panickssery, Samuel Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. *Advances in Neural Information Processing Systems* 37 (2024), 68772–68802.

[14] Yichen Qian, Yongyi He, Rong Zhu, Jintao Huang, Zhijian Ma, Haibin Wang, Yaohua Wang, Xiuyu Sun, Defu Lian, Bolin Ding, et al. 2024. UniDM: a Unified framework for data manipulation with large language models. *Proceedings of Machine Learning and Systems* 6 (2024), 465–482.

[15] Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 645–654.