Converting unstructured interviews to structured data using AI

Prepared By
Our Sci LLC
Rose Fontana, rose@our-sci.net
Adie Pregenzer, adie@our-sci.net
Greg Austic, greg@our-sci.net

Reporting to
National Agricultural Producers Data Cooperative
University of Lincoln, Nebraska

Summary

We used OpenAPI's GPTo in 'Structured Output' mode to test a Large Language Model's (AI) ability to convert raw transcripts of a conversation between a producer and technical assistance provider concerning in-field trials into structured data ("a database") following 3 defined JSON schemas. We tested 20 different prompting strategies ("recipes") and compared them on 3 different test transcripts, and scored them against a human-generated 'gold standard' answer using 2 different scorers (Rose and Adie).

In the highest quality interview, the range of average accuracy scores across the 3 databases was 72%, 78% and 81% correct answers, with the best performing recipe in that same interview yielded 87%, 88%, and 87.5% correct answers respectively.

After analyzing the quality of the AI generated results, our main takeaways were:

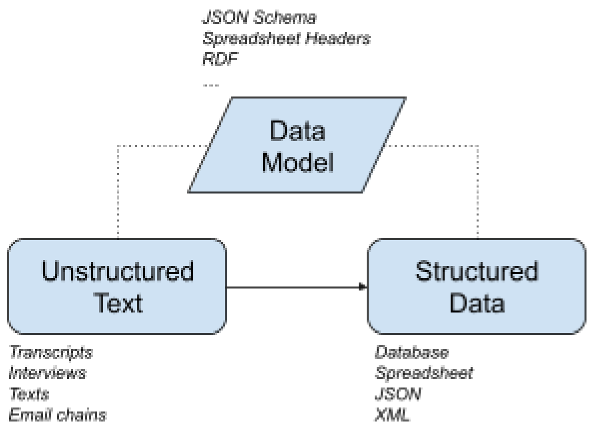
1. the **quality of the interview** was the main driver of the **quality structured output** by the AI;
2. **more complex prompting did not yield clearly better results** (pre-prompts, heavily structured prompts, XML tagging, etc.);
3. **more context and access to history** (field names, interview participant names and roles, etc.) **will probably reduce AI hallucinations** due to missing information;
4. **better defined schemas** with fewer open ended questions will probably result in more accurate results and less hallucinations.

Overall, we feel that with appropriate adjustments based on our learnings, consistent performance of 90% or higher is possible. This exceeds the performance level minimum (ranging from 60 - 90%) identified by the NAPDC interview group during needs assessment.

The Gitlab issue containing complete discussion and results can be found here.

Introduction

Structured outputs is a feature of OpenAI for GPT-4o models that ensures the responses from the model adhere to JSON schema. This kind of a feature is most useful for applications where AI-generated data is pushed into other systems that are structured - like a database. In the domain of agriculture, this means that unstructured data can be given to a chat model and then organized in a structured way. This could be very broadly useful, for example: a farmer might want to organize their inventory by speaking into their phone, locate their equipment using image-based identification, or (most applicable to this group) to convert a conversation (transcript or recording) between a technical assistance provider and a producer into a database of farm management events (seeding or harvest events, on-farm trial details, etc.). These solutions are possible if AI is good at taking unstructured text and converting it into structured data. **In this experiment, we try to determine which AI prompts and parameters result in the most successful conversion from unstructured to structured data.**



The general concept tested - converting unstructured text into structured data using a data model.

Methodology

Experimental Setup

This work focused on varying the prompting, pre-processing, and slight GPT-4o model variation to determine which strategy best achieves our goal of high-quality unstructured to structured data conversion. We call each of these AI strategies a **'recipe'**.

In order to test each recipe, 3 interviews were conducted which followed a single test case. The test case was:

A technical assistance provider is working with a producer to help track data about the producer's in-field trials and learn from the producers' in-field trials. The unstructured data is a single interview in which the technical assistance provider is following up on progress from a trial with the producer.

The 3 interview transcripts (called “Ben Beans”, “Liz Carrots”, and “Wally Squash”) were made intentionally with varying quality and contained different types of complexity like duplicate or incomplete information. These interviews were then converted to interview transcripts using Otter.ai prior to testing. You can find all the information about each of the 3 transcripts in the “Baserow - Input Data” link below.

Lastly, we asked the AI to translate the data from these transcripts into 3 separate structured JSON outputs which had to follow a provided data model in the form of JSON Schema (a JSON Schema here is functionally equivalent to a database with defined data types or, for practical purposes, a spreadsheet with defined column names). The schemas can be found in the “Gitlab - Output Schemas” link below. These 3 schemas were:

1. **Interactions** - Information about the conversation itself - who was present, the summary, and next steps.
2. **Plantings and Fields** - Historical, current, or future information about what farm management activities in the fields mentioned. Seeding, Harvest, Spray events, etc.
3. **Trials** - Information about the in-field trial itself, including what is being compared, independence and dependent variables, and learnings.

Key Links

- Baserow - Input Data
- Baserow - Exported Test Data
- Gitlab - Output Schemas

Recipes and Parameters

A total of 40 recipes were created and of them 20 were tested. The recipes we tested are based on a mix of parameters we came up with internally, along with some suggested “best practices” after light research from Anthropic’s Claude and OpenAI’s ChatGPT websites. Replicate runs were not performed as we did not have time or funding, though we recognize that would improve the quality of the data.

A full description of the parameters can be found in the appendix, while a full list of recipes can be found in the “Baserow - Exported Recipes” link.

Scoring

Scoring was calculated as a % of the structured data types that were correctly marked by the AI (compared to the Gold Standard answer), with additional notes specifying any special types of failures, repeated failures, or other special information. The evaluation criteria for analyzing “correctness” of the structured responses was ultimately subjective, but based on two different considerations deemed most useful in our circumstances.

1. **The data must be valid** according to the data model's requirements - if `boolean` it must be 0 or 1, for example. If it fails this criteria the data will not be accepted by the database or other source and is thus not useful.
2. **The data has to be useful and correct** from an operational management perspective for producer's or TAPs who are going to retrieve the data from the database. In both of these perspectives, the data has to be accurate in that it reflects the reality of operations as the producer understands.

Generating Final Test Runs

The final results were generated in 2 phases. Initially, using the "Bulk" mode, we ran 20 different recipes across 3 interview transcripts and associated data, generating 3 structured data outputs (JSON) each. Scoring was performed by 2 different reviewers (Adie and Rose) and for all runs.

After the initial phase, we found some errors / improvements in our "Gold Standard" outputs which were used to determine the scoring, so we updated it and re-ran 5 recipes to see how much the scoring would change, also performed with the "Bulk" mode. All of the results can be found in the Key Links above under "Baserow - Exported Test Runs".

Key Links

- Baserow - Exported Test Data
- Bulk Results by Recipe
- Appendix

Technical Setup

The technical setup for testing recipes was first to set up a Hugging Face Spaces application. This is a framework for quickly and easily creating ML demos, which comes with features that facilitate this. Gradio is an option for a front-end framework, an open-source Python package that makes it easy to create and connect to machine learning models, APIs and built-in Python functions to serve as middleware for easy connecting. Hugging Face Spaces allows you to swap out machine learning models easily by simply switching an ID or connecting to a different Hugging Face Inference Endpoint, which makes it an attractive choice for experimentation. Hugging Face Spaces also makes it easy to be collaborative, by your project being accessible and running for other users to interact with.

Through this application, we've created two separate modes for operation. The first is a "One Off" mode of operation that is for quick testing and one-off experimentation. The interface is built through Hugging Face, but uses an embedded SurveyStack Survey. SurveyStack is a flexible and collaborative research platform for creating, storing and managing surveys and scripts. The user fills out a survey to set the recipe, picking out each parameter and input data individually that will be used to create the structured outputs for the 3 schemas (Plantings and Fields, Interactions, and Trials and Treatments). The AI generated output can be copied out once completed and viewed.

The second “Bulk” mode of operation uses [Baserow](#) but is still run from Hugging Face. Each recipe is defined in Baserow, including the prompt texts and parameters. An API call is then made to Baserow from the Hugging Face Space to pull the recipe’s and run them all in sequence. It then saves the resulting AI generated JSON outputs in a set of folders.

Key Links

- Hugging Face Space - One Off mode
    - Visible as the embedded SurveyStack survey
    - Ran with “Create JSONs” button
- Hugging Face Space - Bulk mode
    - Visible as a button “bulk
    - Ran with “Generate Output Folder and Download” (downloads zipped folder)

** Note - you may need to ‘restart the space’ on arrival to the above links. It may take a few minutes to initiate the server.*

Results

Raw Score analysis

Scoring of the results were compiled by recipe and compared across the 3 different transcripts. The results were then colored (green, high; red, low) to make comparison faster and easier. The table shows the score x recipe x transcript x output schema, as well as totals across each. The first Scoring Run is separated from the 2nd (in which the Gold Standard results were improved) since the scoring criteria changed between the two.

		Ben Beans					Liz Carrots					Wally Squash					Total Average - Score - Plantings and Fields	Total Average - Score - Interactions	Total Average - Score - Trials	Total Average - Score - Total	Total Count - Score - Total
Scoring Run	Recipe Used	Average - Score - Plantings and Fields	Average - Score - Interactions	Average - Score - Trials	Average - Score - Total	Count - Score - Total	Average - Score - Plantings and Fields	Average - Score - Interactions	Average - Score - Trials	Average - Score - Total	Count - Score - Total	Average - Score - Plantings and Fields	Average - Score - Interactions	Average - Score - Trials	Average - Score - Total	Count - Score - Total					
1	5	65%	88%	88%	77%	2	65%	32%	80%	62%	2	38%	63%	50%	47%	1	59%	60%	77%	65%	5
	7	52%	88%	63%	63%	2	71%	32%	71%	62%	2	69%	63%	50%	61%	1	63%	60%	63%	62%	5
	9	78%	82%	80%	78%	2	52%	25%	59%	47%	2	31%	63%	17%	33%	1	58%	55%	59%	57%	5
	11	69%	69%	71%	69%	2	67%	25%	67%	52%	2	56%	63%	42%	52%	1	65%	50%	63%	58%	5
	13	64%	88%	75%	73%	2	68%	13%	88%	62%	2	50%	69%	50%	58%	1	61%	57%	71%	65%	5
	15	91%	75%	80%	83%	2	78%	44%	71%	67%	2	44%	63%	50%	51%	2	71%	61%	67%	67%	6
	17	87%	88%	88%	87%	2	84%	38%	67%	67%	2	63%	82%	45%	62%	2	78%	69%	67%	72%	6
	19	72%	44%	75%	66%	2	58%	32%	59%	52%	2	50%	44%	46%	48%	2	60%	40%	60%	55%	6
	21	84%	76%	88%	83%	2	23%	32%	54%	35%	2	28%	76%	38%	42%	2	45%	61%	60%	53%	6
	23	78%	88%	59%	73%	2	52%	13%	54%	44%	2	38%	69%	55%	51%	2	56%	57%	56%	56%	6
	25	91%	94%	88%	90%	2	68%	32%	80%	63%	2	44%	69%	33%	47%	2	67%	65%	67%	67%	6
	27	68%	82%	96%	81%	2	81%	26%	75%	66%	2	53%	88%	54%	62%	2	67%	65%	75%	69%	6
	29	65%	76%	88%	75%	2	62%	26%	55%	50%	2	23%	69%	46%	41%	2	50%	57%	63%	55%	6
	31	72%	63%	83%	73%	2	74%	26%	54%	56%	2	49%	56%	67%	56%	2	65%	48%	68%	62%	6
	33	65%	32%	75%	61%	2	61%	19%	71%	54%	2	20%	63%	45%	38%	2	48%	38%	64%	51%	6
	35	85%	88%	100%	90%	2	75%	19%	84%	68%	2	25%	88%	71%	55%	2	61%	65%	85%	71%	6
	36	81%	88%	88%	85%	2	84%	63%	88%	80%	2	38%	63%	17%	36%	1	73%	73%	73%	73%	5
	37	75%	75%	79%	76%	2	45%	50%	88%	61%	2	88%	75%	25%	36%	1	66%	65%	72%	62%	5
	38	58%	76%	54%	61%	2	68%	50%	71%	65%	2	44%	76%	42%	51%	2	56%	67%	56%	59%	6
	40	69%	88%	67%			62%	38%	54%	51%	1	50%	50%	58%			61%	54%	58%	51%	1
Total Result		73%	77%	79%	76%		65%	32%	69%	58%		45%	67%	45%	49%		61%	58%	66%	61%	
2	9	56%	63%	100%	74%	1	74%	63%	75%	71%	1	31%	75%	42%	46%	1	54%	67%	72%	64%	3
	17	68%	75%	100%	83%	1	74%	25%	84%	66%	1	13%	63%	67%	42%	1	52%	54%	84%	64%	3
	27	68%	88%	100%	86%	1	87%	25%	100%	78%	1	38%	75%	50%	51%	1	64%	63%	83%	72%	3
	35	56%	88%	75%	71%	1	54%	50%	75%	60%	1	31%	63%	84%	57%	1	47%	67%	78%	63%	3
	36	63%	75%	100%	80%	1	54%	75%	75%	66%	1	25%	88%	67%	54%	1	47%	79%	81%	67%	3
Total Result		72%	77%	81%	76%		65%	33%	71%	59%		41%	69%	49%	49%		60%	59%	68%	62%	

Based on the raw scores and averages, the best performing transcript was Ben Beans (far left). The output data schema that performed best varied based on the transcript. Liz Carrots -

Interaction scored very poorly relative to the other schemas in Liz Carrots, while Ben Beans - Interaction scored very well relative to the other schemas in Ben Beans.

The top 4 recipes based on overall average score are:

Recipe	Description	Score
36	GPT 4o; No pre-processing; no pre-prompt text; additional background context to set up the situation when requesting the creation of the unstructured data.	73%
17	GPT 4o; No pre-processing; no pre-prompt text; simple 1 sentence prompt when requesting the creation of the unstructured data.	72%
35	Duplicate of Recipe 17	71%
27	Duplicate of Recipe 17	69%

The top 4 recipes based on overall average score for the best performing interview (Ben Beans) are:

Recipe	Description	Score
25	GPT 4o; pre-processing with contextual tagging using GPT 4o; simple 1 sentence prompt when requesting the creation of the unstructured data.	90%
35	(see above)	90%
17	(see above)	87%
36	(see above)	85%

Recipes 35, 17 and 36 were present in both lists. **These best performers were also among the simplest recipes in the list**, including no pre-prompt strategies, no pre-processing, and generally simple prompts when requesting the creation of unstructured data.

Detailed Data Review

An example of one of the structured outputs that are produced from this setup is below, this one in particular is based on an input interview of a producer and agronomist discussing a trial involving carrot varieties. This output below was created from ChatGPT based on a combination of input parameters.

```
{
  "name": "Unknown Field",
  "description": "Carrot variety trial to assess performance.",
  "plantings": [
    {
      "name": "2024 Carrot Variety Trial",
      "status": "archived",
      "crop": [
        "carrot"
      ],
      "variety": [
        "unknown"
      ],
      "logs": [
        {
          "convention": "log--seeding--seeding",
          "date": "2024-06-01",
          "description": "Planted five carrot varieties in the same row for
variety trial."
        },
        {
          "convention": "log--observation",
          "date": "2024-09-01",
          "description": "Harvested carrot varieties for trial and rated based
on yield, appearance, greens, and flavor."
        }
      ],
      "soil": null,
      "yield_": [
        {
          "quantity": null,
          "quality": "Rated on appearance, greens, flavor, and yield."
        }
      ]
    }
  ]
}
```

The detailed data review deep dive was into the *kinds* of errors which were produced during the experiment, which can be found after the appendix, here. The kinds of things that went wrong ranged from the output schemas having “null” unexpectedly when it could have produced data, missing key data that was expected to be pulled, or hallucinations about data which wasn’t provided in the interview at all. There were some cases of additional data being inserted that weren't expected, which in some cases helped to improve the subjective “Gold Standard” by providing a perspective that wasn’t previously considered. In some cases, this additional data was wrong or based on syntax mistakes in the otter.ai transcript of the interview.

In this detailed data review, we can see that there are some kinds of errors which are preferred to others. For example, if the model isn’t certain (and has a low probability of being correct), it should favor the field being “null” or blank as opposed to hallucinating data. These kinds of errors can be fixed more easily and seen in a database.

Discussion + Conclusions

If the more complex and advanced parameters (like using the COSTAR framework, or XML tagging) were more effective, we should have seen some evidence of it. However, we clearly did not. In fact, the consistently highest scoring recipes were the simplest ones, with the exception of recipe 36, which adds some limited situational context. **Conclusion**: This recipe framework will remain important as new parameters and options may appear and models may change to take advantage of other strategies, however, among those tested today with this data it is clear that the simple recipes were most successful.

The clearest differences were across the transcripts themselves, not across the recipes. This indicates that high quality transcripts will result in higher quality outcomes (garbage in, garbage out), regardless of parameters. **Conclusion**: Focus future efforts on improving the quality of the transcript through training or support materials (printed lists of the database to not miss items).

In reviewing specific AI failures, we also identified more hallucination or confusion when the AI was referencing something for which it had no context (existing fields, plantings, the roles of the people in the discussion, etc.). We propose that with more accurate historical context AI will be less likely to hallucinate known facts as well as less likely to confuse relationships between new information and the historical background. The AI could access databases and query them prior to the conversation, using the context prepended to the user's prompt as background information. **Conclusion**: Providing AI access to a database of people/roles and a database of planting/field information will probably help reduce hallucinations and improve translation quality.

Lastly, also in reviewing specific AI failures, it appears the AI is more likely to hallucinate or make mistakes when the data model is unspecific. For example, nearly half of the interactions data model was expecting a date in MM/DD/YYYY format. The interviews were not specific on an actual day for this date, so in the majority of cases, instead of stating null as was expected, the model hallucinated a random day. Also in the interactions schema, there was a concept of a 'role', which the model could pick from a list between "partner", "staff", "agronomist", or "other".

The descriptions for each provided some insight to each role, but based on the output data not enough for the model to appropriately determine which one should be used when. **Conclusion**: Using well described, specific and well separated data models may reduce hallucinations and increase the rate of capturing nuanced information in a usable way.

Questions and Recommendations

Questions

The questions / discussions before proceeding are:

1. Interview quality, and thus interviewer training, is really important. How likely is it that interviewers, or others inputting voice to text data or other types of information, can be trained and will follow training? What are the limits of what they would find useful or reasonable?
2. Investing in high quality schemas will make a difference. We have well defined, high quality schemas around farm management data with the Common Farm Convention. However, we also know that the flexibility to quickly create (for example) a spreadsheet as the data model can make this technology more accessible and flexible. What are the use cases we should focus on (longer-term, high quality CFC, shorter term and more immediately, flexible, but maybe lower quality (?) spreadsheets)?
3. Let's review translations and confirm quality minimums. Are these translations good enough? Are the types of failures acceptable, or particularly bad (making the AI tool not worth it).

Recommendations

Overall, we feel that with appropriate adjustments based on what we've learned and the list of Conclusions above, consistent performance of 90% or higher is possible. This exceeds the performance level minimum (ranging from 60 - 90%) identified by the NAPDC interview group during needs assessment. We feel that continuing progress could achieve real, usable MVPs by the NAPDC cohort (subject also to the questions above). Given that, we recommend:

1. Further investigate Claude or other direct JSON Schema options. If we want a great deal of flexibility, we need to avoid the conversion of JSON Schema → Python classes needed currently to utilize GPT o1's Structured Data Mode. If we avoid this step, the AI solution could be 'plug and play' relative to the data model, allowing for more testing on more data models by more people more easily.
    a. Also add a validation function accessible by Claude.
2. Constrain the interviews - give the TAP a printed, easy to read version of the data model to review during the interview, and teach the interviewer to prompt key pieces of information.
    a. Improve the quality of passed metadata - so the current date, a DB of people + roles, list of fields and plantings from their DB.

3. Improve the schemas or just use Common Farm Convention schemas so the data is specific and well described.
4. Design around specific input SOPs - for example, I want to add new logs / assets through an interview which may reference historical fields and plantings (which can be pulled in from my FMIS). In this case, no editing, no multiple back-and-forths or queries, references are all present or generated on the fly. Or - I want to add a new row to the baserow database and may want to reference another base which is available in context.
5. We should pull from the database for context to populate more accurate context. Specifically, a list of names/roles, a list of fields, and a list of plantings. Also tools for validation

Future Outcomes

Assuming a completion of the recommendations and some level of funding (dependent on exact applications, feedback from the group, and scope), we believe that we can achieve the following:

1. Create at least one real world, usable MVP translating unstructured data to structured data similar to the example cases described above.
2. Assemble the tools and SOPs to make creating new AI unstructured -> structured data MVP applications significantly easier to achieve, faster to evaluate, and lower risk of unexpected failure.

References

https://arxiv.org/pdf/2404.05499
Guiding Large Language Models to Generate Computer-Parsable Content

Appendix

Full Parameter Description

1. **Schema Processing Model**
    a. The schema processing model refers to the model which is used to generate the JSON schema output. We used the most up to date versions of OpenAI's chatGPT which were capable of "JSON mode" structured output. One of the model's was 'gpt-4o-2024-08-06' which is the 'full model' with higher processing capabilities and 'gpt-4o-mini-2024-07-18' a smaller, faster model.
2. **Pre-processing Strategy**
    a. For some of the test recipes, we pre-processed the data prior to creating the JSON schema. There were pre-processing strategies which first processed the data in some manner through an LLM (from the options of the GPT models above). There were also pre-processing strategies which involved either otter.ai or a human summarizing the interaction as input.
    b. Full pre-processing strategies attempted:
        i. Summarization pre-processing (through GPT model):
            1. You are given a block of text containing detailed farm management data, interactions, or trial and treatment descriptions. Your task is to generate a concise summary of the text, focusing on the main points and removing unnecessary details. Ensure that the summary captures the essential information that will be useful for the next model to process it into structured data.
                a. The summary should include:
                    - The primary field or subject (e.g., farm name, interaction, trial name)
                    - Key attributes or details (e.g., crops, treatments, soil, participants)
                    - Any specific action or outcome (e.g., meeting dates, crop yield, next steps)

                Keep the summary brief and informative, maintaining the most important information for further processing.

        ii. Otter.ai summary (provided from otter.ai interface automatically)
        iii. Greg summary (interaction was summarized by the human Greg as input)
        iv. **Specific Field Extraction:**
            1. You are given a block of text containing detailed information. Your task is to extract and structure the following specific fields:
                a. For **Farm Management Data**:
                    - **Field Name**: The name of the field.

- **Description**: Any description of the field (e.g., planting year, soil conditions).
- **Crops**: The types of crops planted (e.g., squash, soybeans).
- **Soil Conditions**: The type of soil described (e.g., loam, sandy).
- **Yield**: The quantity and quality of the yield (e.g., average, some early rotting).

For **Interaction Data**:

- **Date**: The date of the interaction.
- **Next Meeting**: Any future meeting dates or ranges.
- **Next Steps**: The follow-up actions mentioned.
- **Summary**: A brief summary of the interaction.
- **People**: Names and roles of people involved.

For **Trial and Treatment Data**:

- **Trial Name**: The name of the trial.
- **Description**: A description of the trial.
- **Treatments**: The treatments used in the trial.
- **Crops**: The crops involved in the trial.

Please extract these fields and return them in a structured format. If any fields are missing, leave them blank.

v. **Contextual Tagging:**

You are given a block of text. Your task is to tag each part of the text with appropriate labels, such as:

- **Location**: Tag locations where mentioned.
- **Date**: Tag dates or time ranges.
- **Action**: Tag actions or verbs that indicate activity (e.g., "planting", "harvesting").

Return the text with these labels inserted, marking the relevant parts clearly.

c. **Prompting Strategy:**

i. The last parameter we tested was prompting strategies, which was similar to our pre-processing prompting strategy, but included a wider range of options. We also had to ensure we had different prompts for each of the schemas, which are slightly tweaked for the differences in wording between the Plantings and Fields Schema, Interactions Schema, and Treatments Schema.:

1. Baseline Approach (Default)
    a. Plantings and Fields:
        i. Please extract the farm management data.
    b. Interactions:
        i. Please extract the interactions and people data.
    c. Treatments:
        i. Please extract the trials and treatment data.
2. **Baseline Approach - Additional Context**
    a. Plantings and Fields:
        i. The following is a written transcript of an interaction between a Technical Assistance Provider (TAP) and a producer. The people in this interaction are discussing an experiment. You will be creating structured JSON output based on the discussion of the experiment, which should result in a structured JSON schema for the fields and plantings discussed. This JSON schema should include extracted details like the field name, description of the field, the types of crops and plantings and soil conditions.
    b. Interactions:
        i. The following is a written transcript of an interaction between a Technical Assistance Provider (TAP) and a producer. The people in this interaction are discussing an experiment. You will be creating structured JSON output based on the discussion of the experiment, which should result in a structured JSON schema for the interaction discussed. This JSON schema should include extracted details like the meeting date, the date of the next meeting, a list of people involved in the interaction along with their roles, and a summary of the interaction.
    c. Treatments:
        i. The following is a written transcript of an interaction between a Technical Assistance Provider (TAP) and a producer. The people in this interaction are discussing an experiment. You will be creating structured JSON output based on the discussion of the experiment, which should result in a structured JSON schema for the trials and treatments discussed. This JSON schema should include extracted details like the name of the trial, a

description of the trial, the treatments that are being tested in the trial, and a list of the crops involved in each treatment.

3. **Full Additional Context - Rules and Constraints**
    a. Plantings and Fields:

    The following is a written transcript of an interaction between a Technical Assistance Provider (TAP) and a producer. The people in this interaction are discussing an experiment. You will be creating structured JSON output based on the discussion of the experiment, which should result in a structured JSON schema for the fields and plantings discussed. This JSON schema should include extracted details like the field name, description of the field, the types of crops and plantings and soil conditions.

    Rules and Constraints:

    1. There should be a separate Planting for each FarmActivities if there is a different crop type, day or time or crop variety. This is an account of each time a Planting occurs as an activity.
    2. Each planting must have a Planting Log with the convention = "log--activity--planting".
    3. For each application of an Input like herbicides, irrigation, or fertilizers are discussed, create Input logs with the appropriate conventions:
    Herbicide/Pesticide:
    log--input--herbicide_or_pesticide
    Irrigation: log--input--irrigation
    Organic Matter: log--input--organic_matter
    4. Tillage Log: If tillage is mentioned, create a log with convention = "log--activity--tillage"
    5. Irrigation Log: If irrigation is discussed, create a log with convention = "log--input--irrigation"
    6. Solarization Log: For solarization, use convention = "log--activity--solarization"

    b. Interactions:

    The following is a written transcript of an interaction between a Technical Assistance Provider (TAP) and a producer. The people in this interaction are discussing an experiment. You will be creating structured JSON output based on the discussion of the experiment, which should result in a structured JSON schema for the interaction discussed. This JSON schema should include extracted details like the meeting date, the date of the next meeting, a list of people involved in the interaction along with their roles, and a summary of the interaction.

Rules and Constraints:

1. There should be a separate Person for each different Person mentioned during the interaction who is involved with the current interaction.
2. The nextMeeting of the Interactions should be the date that is scheduled for the producer to next meet with the TAP. If none is explicitly provided, the nextMeeting date should be null.
3. The nextMeeting of the Interactions should occur after the current meeting date.
4. The nextSteps should be a list of individual steps that the people involved in the interaction will have to take to complete the trial which will occur after the interview.
5. The summary should be everything that occurred in the interaction, taking in account special attention to the field and plantings and treatments and outcomes of the trial taking place.

c. Treatments:

The following is a written transcript of an interaction between a Technical Assistance Provider (TAP) and a producer. The people in this interaction are discussing an experiment. You will be creating structured JSON output based on the discussion of the experiment, which should result in a structured JSON schema for the trials and treatments discussed. This JSON schema should include extracted details like the name of the trial, a description of the trial, the treatments that are being tested in the trial, and a list of the crops involved in each treatment.

Rules and Constraints:

1. For each independent variable which is being tested in the trial, there should be a Treatment.
2. Each Treatment should contain a list of crops and fields involved in the specific variables being tested in the Treatment. For example, if the amount of irrigation is being tested, there should be a

Treatment with Variables accounting for the controlled variables, which are unchanging, and the independent variable which is the one specifically being tested in the Treatment.

3. Each Variables outcome should be the projected hypothesized outcome based on the what the people theorized would occur during the interaction, if explicitly mentioned.

4. **Example-Driven**
   a. Plantings and Fields:
      i. Prompt:
         "Please extract farm management data, including field name, description, crop types, and soil conditions in a structured JSON format."
      ii. Example Input:
         "North Field, planted in 2016, grew acorn squash in 2023. Soil: loam, sandy in spots. Yield: average, some early rotting."
      iii. Example Output (JSON):
      iv. {
         "name": "North Field",
         "description": "Planted in 2016",
         "plantings": {
         "name": "2023 Squash",
         "crop": ["squash"],
         "variety": ["acorn squash"],
         "soil": {
         "structure": ["loam", "sand"]
         },
         "yield": {
         "quantity": "average",
         "quality": "some early rotting"
         }
         }
         }
   b. Interactions:
      i. Prompt:
         "Please extract interaction data including date, participants, next meeting, next steps, and a summary in JSON format."
      ii. Example Input:
         "Meeting with Ben Austic on 2023-10-25. Next meeting: 2023-11-15 to 2023-12-25. Discuss soybean trial results and follow up with Ben."

iii. Example Output (JSON):

iv. {
"date": "2023-10-25",
"nextMeeting": "2023-11-15 to 2023-12-25",
"nextSteps": [
"Follow up with Ben Austic to evaluate soybean trial."
],
"summary": "Meeting with Ben Austic to discuss soybean trial results.",
"people": {
"name": "Ben Austic",
"role": "partner"
}
}

c. Treatments:

i. Prompt:
"Please extract trial and treatment data including trial name, description, treatments, and crops in JSON format."

ii. Example Input:
"Ben's soybean trial compares two seedling treatments: fungicide, herbicide, and biologicals for early immune boost. Trial includes soybeans."

iii. Example Output (JSON):

iv. {
"name": "Ben's Soybean Seedling Trial",
"description": "Comparing two seedling treatments with fungicide, herbicide, and biologicals.",
"treatments": {
"name": "Seedling Treatment",
"description": "Includes fungicide, herbicide, biologicals for immune boost.",
"crops": ["Soybeans"]
}
}

5. Step-by-Step Instructional

a. Plantings:

i. Please extract the farm management data. Identify the activity mentioned (e.g., planting, irrigation, harvest, etc.).
Extract the date of the activity.
Note the weather conditions during the activity.
Identify the soil type or field conditions mentioned.

Summarize the activity, including all of the details above.

b. Interactions:
    i. Please extract the interactions and people data.
    Identify the people mentioned in the text and their roles (e.g., partner, agronomist, staff).
    Note the date of the interaction.
    Summarize the content of the interaction (what was discussed or decided).
    Extract any next steps or follow-up tasks mentioned in the text.
    Return the information in a structured format with names, roles, and any tasks or meetings scheduled.

c. Treatments:
    i. Please extract the trials and treatments data.
    Identify the trial name and description.
    Extract the date and treatments applied (e.g., herbicide, irrigation).
    List the crops and fields involved in the treatment.
    Note any expected outcomes or observations related to the treatment.
    Return the extracted details in a structured format including the trial name, treatments, crops, and expected outcomes.

6. **Role-Specific (Agronomist)**
    a. Plantings and Fields:
        i. As an agronomist, you need to extract farm management data. Focus on operational activities and key practices. For each activity, include the crop types involved, the date, the specific action (e.g., seeding, irrigation), and the status of the activity (e.g., ongoing, completed). Capture any relevant observations, such as soil conditions or pest presence, and organize the data into a structured JSON format.
    b. Interactions:
        i. As an agronomist, extract data related to interactions between farm staff. Focus on key meetings, collaborations, and future tasks. Include names of people involved, their roles, and any important follow-up actions. Format the data into a structured JSON format to capture the relevant details.

c. Treatments:
    i. As an agronomist, you are responsible for extracting trials and treatment data. Focus on ongoing field trials, including the treatments used, the crops involved, and the objectives of the trial. Ensure to capture the treatment details, including any chemicals or biological agents, and their effects on crop growth. Present the data in a structured JSON format for easy analysis.

7. **Role-Specific (Data Scientist)**
    a. Plantings and Fields:
        i. As a data scientist, your task is to extract farm management data and format it for analysis. Focus on identifying key operational activities, such as seeding or irrigation, and provide data on the crop types involved, date, and the status of each activity. Ensure that the data is organized in a way that can be easily analyzed for trends and decision-making.
    b. Interactions:
        i. As a data scientist, you will extract interaction data between farm staff to help analyze team dynamics and decision-making. Capture details on meetings, the roles of attendees, and any action items discussed. The data should be structured in a JSON format for easy integration into workflow analysis tools
    c. Treatments:
        i. As a data scientist, you need to extract and analyze trial and treatment data. Focus on ongoing trials, capturing the types of treatments used, the crops involved, and the desired outcomes. Ensure that you gather detailed information on all experimental factors, including treatment variations, to facilitate a comprehensive analysis of their effectiveness.

8. **Error Detection:**
    a. Plantings and Fields:
        i. Please extract the farm management data.
        If any important detail (such as the date, activity type, or crop) is missing from the text, flag it as incomplete.
        Also, check that the activity has a clear description.
        If there are discrepancies in soil types or field descriptions, indicate them as potential errors for review.

b. Interactions:
    i. Please extract the interactions and people data.
    If the roles or names of the participants are missing or unclear, flag them as incomplete.
    Verify that the date and next steps are clearly mentioned; if they are absent or ambiguous, highlight them for review.
c. Trials and Treatments:
    i. Please extract the trials and treatments data.
    Ensure that the trial name and description are complete. If the treatment type, date, or expected outcomes are missing or unclear, flag them as errors.
    If the crops or fields involved are not specified, mark it as incomplete.

9. CO-STAR Framework

a. Plantings and Fields:

CONTEXT

You are tasked with extracting farm management data, including details about activities such as planting, irrigation, and harvesting, as well as information on crops, soil conditions, and weather. This data is used for farm reporting and analysis.

##############

OBJECTIVE

Your goal is to extract and structure farm management data:

- Identify activities like seeding, irrigation, and harvesting.
- Extract the date of each activity.
- Note the crops involved in the activity.
- Extract information regarding soil conditions, structure, and biology.
- Capture any relevant weather information during the activity.
- Summarize yield information, including quantity and quality.

##############

STYLE

Provide the data in a structured format, ensuring that all relevant details are captured concisely and without unnecessary elaboration. Focus on accuracy and clarity.

##############

TONE

The tone should be neutral, focusing on providing an accurate and objective summary of farm activities and conditions.

##############

AUDIENCE

This data is intended for agronomists, farm managers, and data analysts who need a structured overview of farm activities for reporting and decision-making.

##############

RESPONSE

Provide the extracted data in JSON format, following the provided schema for farm management. Ensure that all relevant fields are populated and the data is easy to interpret.

b. Interactions:

CONTEXT

You are tasked with extracting interactions and people data from text that describes farm-related meetings, collaborations, and follow-up actions. This includes identifying people, their roles, and summarizing what was discussed or decided during the interaction.

##############

OBJECTIVE

Your goal is to extract and structure interactions and people data:

- Identify the names of people involved in the interaction.
- Extract the roles of the people mentioned (e.g., partner, agronomist, staff).
- Note the date of the interaction or meeting.
- Summarize the content of the interaction, such as what was discussed, decided, or planned.
- Identify any next steps or follow-up tasks that were agreed upon.

##############

STYLE

The style should be factual and concise, capturing the key details of the interaction without unnecessary elaboration. Focus on the essential points of the discussion and follow-up actions.

##############

TONE

The tone should remain neutral and objective, focusing on providing an accurate summary of the interaction and any planned follow-up actions.

##############

AUDIENCE

This data is intended for farm managers, team leads, or other personnel tracking team interactions and ensuring follow-up on tasks or meetings.

##############

RESPONSE

Provide the extracted interaction and people data in JSON format, following the provided schema. Ensure that all fields are filled accurately and the information is easy to interpret.

c. Treatments:

CONTEXT

You are tasked with extracting data related to agricultural trials and treatments, including the details of the trials, the treatments applied, and the crops involved. This data is used to monitor and analyze the effectiveness of different treatments on crop growth.

##############

OBJECTIVE

Your goal is to extract and structure trial and treatment data:

- Identify the name and description of the trial.
- Extract the date of the trial and any treatments applied (e.g., herbicides, biologicals).
- Note the crops and fields involved in the trial.
- Capture any expected outcomes or observations related to the trial and treatments.

##############

STYLE

The style should be clear and focused, with descriptions that are concise and to the point. Ensure that all key details are captured in a structured manner.

##############

TONE

The tone should be objective and technical, providing an accurate and neutral report of the trials, treatments, and expected outcomes.

##############

AUDIENCE

This data is intended for agricultural researchers, agronomists, and farm managers who are tracking the progress and outcomes of field trials.

##############

RESPONSE

Return the extracted trial and treatment data in JSON format, adhering to the provided schema. Ensure all relevant fields are populated and the data is clearly structured.