Leveraging LLMs for scalable seismic metadata extraction from unstructured SEG-Y text headers

Edward Tian, Altay Sansal, Alejandro Valenciano, Ben Lasscock, Santhkumar Rajendran, TGS*

Summary

Extracting structured metadata from unstructured SEG-Y text headers is essential for organizing and retrieving seismic data. We develop and assess an LLM-powered API that organizes unstructured SEG-Y text header data using predefined JSON schemas. Across 21 datasets, our method achieves a semantic accuracy of 90.33% and a strict (worst-case) accuracy of 79.45%, demonstrating its effectiveness for scalable data extraction. We explore areas for improvement, including domain knowledge integration and model enhancements, to further optimize structured text extraction.

Introduction

Extracting structured information from unstructured text remains a fundamental challenge in data processing pipelines. Traditional rule-based approaches, such as parsers based on regular expressions, often struggle with variability in text formats, leading to inconsistencies and data loss. Recent advances in Large Language Models (LLMs) have opened new possibilities for utilizing contextual understanding to enhance extraction performance (Moundas et al. 2024).

A significant application of this technology is the curation of seismic data (SEG-Y) files. These files contain a textual header (typically encoded as EBCDIC) that lacks enforced standardization, except for the requirement that the header must consist of 40 lines, each with 80 characters. The header text typically includes metadata pertinent to the seismic dataset, such as the seismic product name, survey date, and coordinate reference system. Traditional extraction methods necessitate manual effort or inflexible rule-based pipelines, both of which are inefficient and prone to error.

We present a workflow API that utilizes GPT-4o (OpenAI 2023) to extract structured metadata from unstructured seismic text headers. Our API formats header text into structured JSON using a predefined schema and enforces consistency through schema constraints. We evaluate extraction accuracy using a golden dataset consisting of human-curated metadata from 21 seismic headers. Our contributions are as follows:

- We develop an LLM-powered API for structured extraction from seismic EBCDIC headers.
- We define a rigorous evaluation framework based on strict and semantic accuracy metrics.
- We analyze performance across 21 datasets and discuss improvements for future iterations.

Method

The API was built using Python, with core technologies including FastAPI, Pydantic, and the OpenAI API library. Unstructured text headers were processed within a QA pipeline. The API request sends header text and a dataset type (e.g., 3D pre-stack data), which determines the expected structure of the extracted output. This dataset type corresponds to a predefined JSON schema via Pydantic, ensuring structured extraction. To enforce consistency, the Pydantic schema is embedded directly into the LLM prompt, guiding GPT-4o to generate structured JSON responses. This structured approach prevents data inconsistencies and ensures that extracted metadata meets expected field constraints. For example, fields such as geodetic datum are validated against known coordinate reference systems, while numeric values are constrained within predefined data types (e.g., float, integer).

Evaluation and Metrics

To assess model variability, 21 seismic text headers with human-curated text extractions were used as a golden dataset. Each header was evaluated 10 times, and extraction results were analyzed using two accuracy metrics.

Strict Accuracy: Measures exact matches between extracted fields and expected outputs. Minor variations (e.g., "Meters" vs. "meters") result in mismatches, making this a rigid evaluation metric.

Semantic Accuracy: Uses an LLM to assess conceptual equivalence in text-based fields while enforcing exact comparisons for numeric fields. For example, “NAD83” and “North American Datum 1983” are considered semantically equivalent and are correct.

By combining these two metrics, we obtain lower and upper bounds for the true accuracy of the extraction API.

Examples

```
C 1 CLIENT: ROCKY MOUNTAIN OILFIELD TESTING CENTER
C 2 PROJECT: NAVAL PETROLEUM RESERVE #3 (TEAPOT DOME);
NATRONA COUNTY, WYOMING
C 3 LINE: 3D
C 4
C 5 THIS IS THE FILTERED POST STACK MIGRATION
C 6
C 7 INLINE 1, XLINE 1: X COORDINATE: 788937  Y COORDINATE: 938845
C 8 INLINE 1, XLINE 188:X COORDINATE: 809501 Y COORDINATE: 939333
C 9 INLINE 188, XLINE 1: X COORDINATE: 788039  Y COORDINATE: 976674
C10 INLINE NUMBER:    MIN: 1  MAX: 345  TOTAL: 345
C11 CROSSLINE NUMBER: MIN: 1  MAX: 188  TOTAL: 188
C12 TOTAL NUMBER OF CDPS: 64860   BIN DIMENSION: 110\' X 110\'
C13
...
C18
C19 GENERAL SEGY INFORMATION
C20 RECORD LENGHT (MS): 3000
C21 SAMPLE RATE (MS): 2.0
C22 DATA FORMAT: 4 BYTE IBM FLOATING POINT
C23 BYTES  13- 16: CROSSLINE NUMBER (TRACE)
C24 BYTES  17- 20: INLINE NUMBER (LINE)
C25 BYTES  81- 84: CDP_X COORD
C26 BYTES  85- 88: CDP_Y COORD
C27 BYTES 181-184: INLINE NUMBER (LINE)
C28 BYTES 185-188: CROSSLINE NUMBER (TRACE)
C29 BYTES 189-192: CDP_X COORD
C30 BYTES 193-196: CDP_Y COORD
C31
…
C35
C36 Processed by: Excel Geophysical Services, Inc.
C37             8301 East Prentice Ave. Ste. 402
C38             Englewood, Colorado 80111
C39             (voice) 303.694.9629 (fax) 303.771.1646
C40 END EBCDIC
```

Figure 1: An example of a seismic EBCDIC header file

```
C 1 SEGY OUTPUT FROM Petrel 2021.4 Wednesday, April 19 2023 09:18:55
C 2 Name: WSA-320 Type: 2D seismic
C 3
C 4 First CDP: 8.000000 Last CDP: 16160.000000
C 5 First SP:  1.000000 Last SP:  803.000000
C 6 CRS: UTM83-5 ("MENTOR:UTM83-5:NAD83 UTM, Zone 5 North, Meter")
NSIS,502402e
C 7 X min: 279550.79 max: 401465.89 delta: 121915.10
C 8 Y min: 6147062.65 max: 6307028.10 delta: 159965.45
C 9 Time min: -6298.00 max: 2.00 delta: 6300.00
C10 Lat min: ~55.42010880 max: ~56.89648136 delta: ~1.47637257
C11 Long min: ~-156.61579948 max: ~-154.55829079 delta: ~2.05750869
C12 Trace min: -6296.00 max: 0.00 delta: 6296.00
C13 Seismic (template) min: ~-29577.00 max: ~30495.00 delta: ~60072.00
C14 Amplitude (data) min: ~-29577.00 max: ~30495.00 delta: ~60072.00
C15 Trace sample format: IEEE floating point
C16 Coordinate scale factor: 100.00000
C17
C18 Binary header locations:
C19 Sample interval           : bytes 17-18
C20 Number of samples per trace : bytes 21-22
C21 Trace date format         : bytes 25-26
C22
C23 Trace header locations:
C24 Inline number             : bytes 5-8
C25 SP Number                 : bytes 17-20
C26 CDP number                : bytes 21-24
C27 Coordinate scale factor   : bytes 71-72
C28 X coordinate              : bytes 73-76
C29 Y coordinate              : bytes 77-80
C30 Trace start time/depth    : bytes 109-110
C31 Number of samples per trace : bytes 115-116
C32 Sample interval           : bytes 117-118
C33
…
C39
C40 END EBCDIC
```

Figure 2: A second example of a seismic EBCDIC header file to illustrate the difference in structure.

Above are two examples of public domain seismic text headers (Fig. 1 & 2). The texts have been edited for clarity. They illustrate the differences in formatting, structure, and terminology. The API's extraction schema is displayed in Figure 3. Every field the API will try to extract can have a description and a data type (e.g., integer), which the LLM will consider when performing the extraction. Additional constraints and information can be injected in the Pydantic class if a field is more complicated or requires additional domain knowledge. Note that for the first iteration of this API, all the descriptions are sparse and provide little additional domain knowledge for the LLM to utilize.

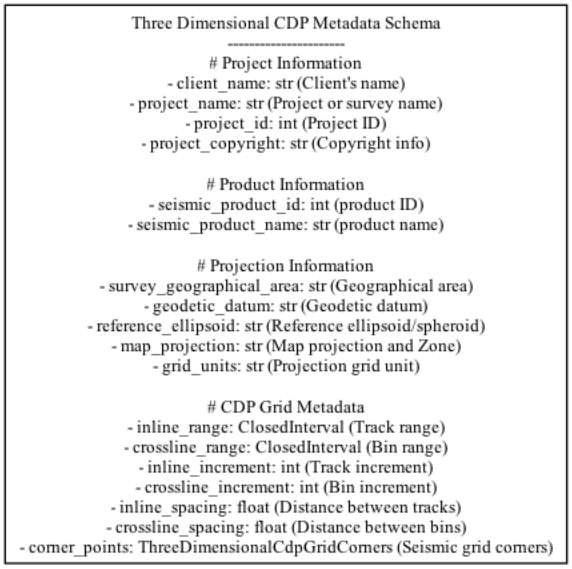

Figure 3: A conceptual model of one of the extraction schemas in the API

Lastly, we have example extraction outputs (Figures 4 & 5) derived from the unstructured text in Figures 1 and 2. The API will not try to fill in missing fields from the header file. However, note that the API does not simply pull text from the header file. For example, in Figure 4, the API deduced that the grid units were in feet, but that is not strictly specified anywhere in the header file; rather, the API most likely inferred this from the bin dimension being 110’ x 110’.

Both examples show that extracting these structured fields from a human perspective is not particularly difficult, but the time-consuming nature of reading through the header file, as well as the large number of header files to process, makes this a good use case for the LLM extraction API.

```
{
    "clientName": "ROCKY MOUNTAIN OILFIELD TESTING CENTER",
    "projectName": "NAVAL PETROLEUM RESERVE #3 (TEAPOT DOME); NATRONA COUNTY, WYOMING",
    "projectId": null,
    "projectCopyright": null,
    "seismicProductId": null,
    "seismicProductName": null,
    "surveyGeographicalArea": null,
    "geodeticDatum": null,
    "referenceEllipsoid": null,
    "mapProjection": null,
    "gridUnits": "feet",
    "inlineRange": {
      "start": 1,
      "stop": 345
    },
    "crosslineRange": {
      "start": 1,
      "stop": 188
    },
    "inlineIncrement": 1,
    "crosslineIncrement": 1,
    "inlineSpacing": 110.0,
    "crosslineSpacing": 110.0,
    "cornerPoints": {
      "corners": [
        {
          "x": 788937.0,
          "y": 938845.0,
          "inline": 1,
          "crossline": 1
        },
        {
          "x": 809501.0,
          "y": 939333.0,
          "inline": 1,
          "crossline": 188
        },
        {
          "x": 788039.0,
          "y": 976674.0,
          "inline": 188,
          "crossline": 1
        }
      ]
    }
  }
```


Figure 4: The extracted output from the header in Figure 1


```
{
    "clientName": null,
    "projectName": "WSA-320",
    "projectId": null,
    "projectCopyright": null,
    "seismicProductId": null,
    "seismicProductName": null,
    "surveyGeographicalArea": null,
    "geodeticDatum": "NAD83",
    "referenceEllipsoid": null,
    "mapProjection": "UTM, Zone 5 North",
    "gridUnits": "Meter",
    "lineNo": "WSA-320",
    "shotpointRange": {
      "start": 1,
      "stop": 803
    },
    "cdpRange": {
      "start": 8,
      "stop": 16160
    },
    "cdpIncrement": null
  }
```


Figure 5: The extracted output from the header in Figure 2

Results

Across 210 API calls (21 headers × 10 runs each), the model attained a strict accuracy of 79.45% and a semantic accuracy of 90.33%. To date, the API has been utilized to augment human-performed quality assurance on over 25,000 SEG-Y files.

Accuracy Distribution Analysis

Figure 6 shows the histogram of accuracy results across all samples. The distribution is left-skewed, indicating that most samples yielded high accuracy, while a few outliers contributed to lower average accuracy. These lower-performing samples often exhibited inconsistent text formatting or required domain-specific knowledge beyond the LLM's pretraining.

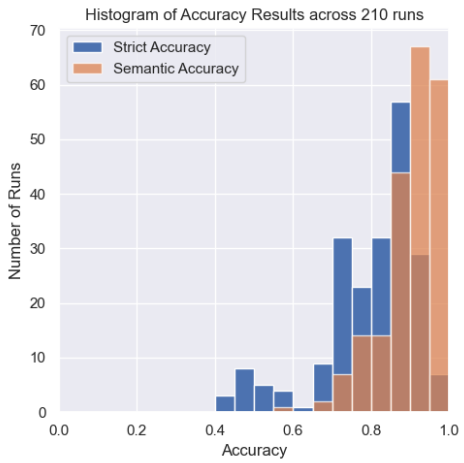

Figure 6: Histogram of accuracy across 210 outputs (21 samples, 10 extracts per sample) for both strict accuracy (in blue) and semantic accuracy (in orange).

File-Specific Extraction Performance

We also observed a strong inverse relationship between extraction variability and accuracy, as illustrated in Figure 7—fields with higher standard deviation showed lower semantic accuracy. This indicates that increasing schema constraints and integrating domain-specific guidance in the API prompt could improve extraction consistency. Furthermore, the extraction variability over multiple runs could serve as a confidence indicator.

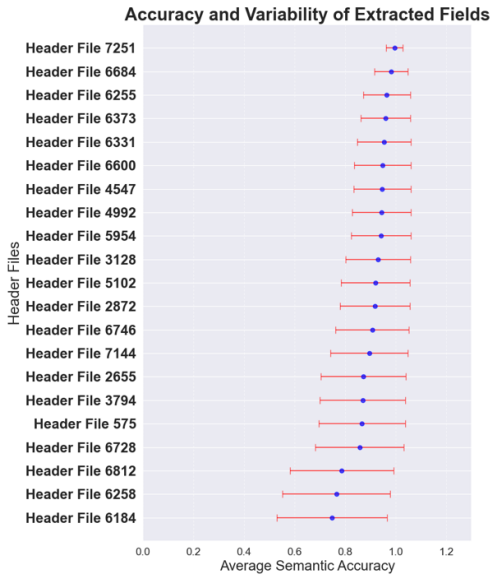

Figure 7: The average and standard deviation of the semantic accuracy of extraction for each header file in the test dataset, sorted by the standard deviation column (scaled down by a factor of 0.5 for readability).

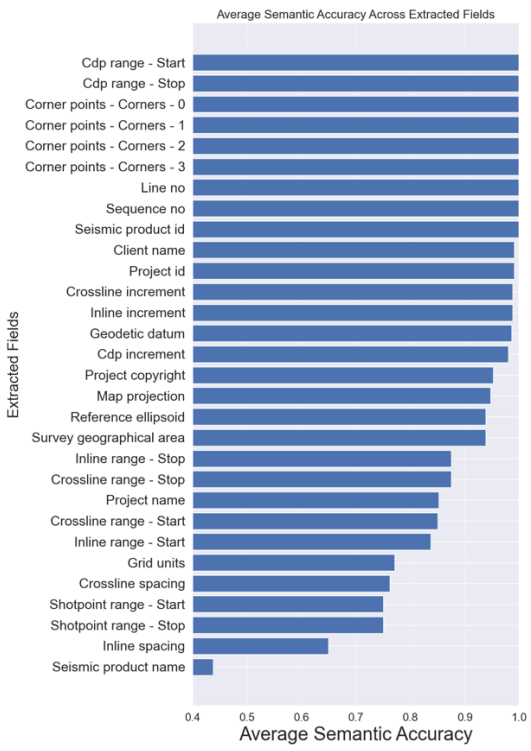

Figure 8: Extract fields sorted by their semantic accuracy.

Field-Specific Extraction Performance

Figure 7 presents a breakdown of extracted fields sorted by their semantic accuracy. Fields such as project ID, client name, and geodetic datum performed well, achieving over 90% accuracy. Other fields that are typically distinct in the header text, like the corner points (which are usually in a tabular format that stands out from the rest of the header file), are consistently extracted accurately. In contrast, fields that require more domain knowledge to distinguish, such as seismic product name and project name, exhibit greater extraction difficulty, with semantic accuracy rates of 60% and 80%, respectively.

Discussion

Key Observations and Future Improvements

This study highlights the potential of LLM-powered structured extraction but also identifies several areas for improvement:

1. **Refining Schema Definitions:** The current JSON schema definitions lack detailed domain knowledge. Expanding field descriptions and adding contextual hints (e.g., defining expected units or formats) could improve extraction consistency.
2. **Few-Shot Prompting (Brown et al, 2020):** The API could integrate historical extractions as examples within the prompt, guiding the model towards more accurate outputs. This would be particularly beneficial for fields with ambiguous formatting.
3. **Multi-Pass Extraction with Consensus Aggregation:** Given the probabilistic nature of LLM outputs, running multiple extractions per header and selecting the most frequently generated value could enhance reliability.

Conclusions

LLMs provide a scalable, adaptable, and efficient solution for extracting structured data from unstructured text. Our study demonstrates that a pre-trained LLM-powered API can effectively extract key metadata fields from seismic textual headers with high accuracy. By evaluating performance across 21 datasets, we highlight the strengths and challenges of applying LLMs to structured extraction tasks. While current performance is promising, we see several opportunities for improvement, including schema refinement, few-shot prompting, and multi-pass extraction. Future research will examine these enhancements to increase extraction accuracy and consistency further.