UNSTRUCTURED DATA PROCESSING: ASKING NEW QUESTIONS OF AVALANCHE ACCIDENT REPORTS AND OTHER NATURAL LANGUAGE DOCUMENTS

Bhavya Chopra[1], Björn Hartmann*[1], Aditya Parameswaran[1], Shreya Shankar[1]

[1]*Electrical Engineering and Computer Sciences Department, UC Berkeley, CA, USA*

ABSTRACT: Statistical analyses are powerful tools for distilling patterns in avalanche accidents. Most such analyses rely on structured data — quantities that can be easily counted or measured — such as forecast danger ratings, burial times, or group sizes. But much of what the avalanche community records lives in unstructured text: accident narratives, crowd-sourced observation reports, and forecast discussions. Finding patterns in such sources has traditionally required manual search and review, which is time-intensive and limits the number of questions one can practically pursue.

Recent advances in large language models (LLMs) have given rise to unstructured data processing: the ability to query and transform large collections of text documents using natural language prompts to define the analysis operations. Rather than searching for keywords, a researcher might ask: "Find all accident reports where skier behavior deviated from the stated plans, categorize the reasons and provide counts." Unstructured data processing can be used both for deductive, hypothesis-driven analysis and inductively, to surface new themes or patterns that emerge from the data itself. We demonstrate this approach on data sets of avalanche accident and incident reports from the United States and Canada.

LLM-powered unstructured data processing is no panacea. LLMs can hallucinate, and processing pipelines can be sensitive to small changes in prompt phrasing or pipeline structure, making results brittle without careful review. We discuss validation strategies to help ensure findings reflect reality. We introduce tooling, including the DocETL system developed at UC Berkeley, that can make unstructured data processing accessible to researchers and practitioners.

KEYWORDS: large language models, unstructured data processing, visualization, avalanche reports

1. INTRODUCTION

Statistical analyses of avalanche incidents are largely built on structured data fields—quantities that can be easily counted or measured—such as forecast danger ratings, burial times, or group sizes. These analyses have shaped how the community understands risk and have informed avalanche education for decades (Gallagher 1967; Armstrong 1980; Logan and Greene 2014; Techel et al. 2016).

Yet, much of the rich, qualitative information about avalanche activities and incidents is locked away in unstructured text-heavy documents, such as accident narratives, crowd-sourced observation reports, forecast discussions, and exchanges in online forums. Manual analyses of these accident narratives have produced influential insights. Atkins' work on human factors drew on careful qualitative review of historical accident records (Atkins 2000). Similarly, McCammon's identification of heuristic traps in avalanche decision-making (McCammon 2002; McCammon 2004) was based on reviewing hundreds of report narratives and coding them into countable categories. Such studies require substantial human effort and experience.

Reading, coding, and categorizing narrative reports is time-intensive and limits the number of questions a researcher can practically pursue. Computational approaches have offered partial relief—for instance, keyword searches can locate reports mentioning specific terms, and tools can perform narrow tasks like sentiment classification. But performing qualitative analyses that require semantic interpretation and triangulation of patterns across reports has remained a manual task, requiring either dedicated domain-expert time or organized crowdsourcing efforts.

Large language models (LLMs) now offer a way forward by enabling unstructured (or semantic) data processing, which allows analysts to query and transform large collections of text documents using natural language prompts. Consider an analyst who wants to identify deviations in skiers' plans that led to accidents. Instead of manually reading each narrative, the analyst can write a prompt: *"For each accident report, identify and summarize where skier behavior deviated from their stated*

* *Corresponding author address:*
Björn Hartmann, EECS Department,
University of California, Berkeley, CA 94720;
email: bjoern@eecs.berkeley.edu

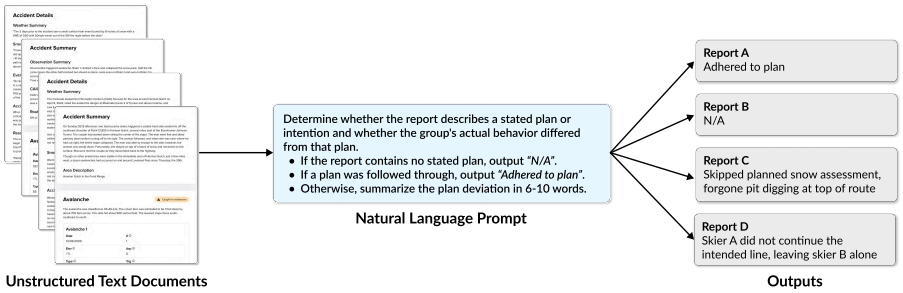


Figure 1: **An unstructured data processing operator.** The LLM-powered operator takes the avalanche accident report collection as input, and uses the specified prompt to produce one output per report, in this case summarizing any plan deviations.

plans." This prompt is passed to an LLM along with one accident report at a time to obtain output summaries (Figure 1). The analyst can now chain additional operations to gather and produce counts of common deviation patterns. With this approach, the same corpus of text documents can be queried with dozens of different questions in a single session, each taking a few minutes to specify and run, at a cost of a few dollars per analysis.

Researchers in qualitative methods have investigated how well such LLM-powered processing can transform unstructured textual corpora from surveys, news and other document collections into structured data (Mellon et al. 2024; Than et al. 2025; Hayes 2025). The promising results suggest that these advancements are likely to transform data analysis in a variety of fields including sociology, political science and the digital humanities. We hypothesize that similar benefits can be realized for the avalanche community.

In this paper, we introduce key concepts behind unstructured data processing, describe how researchers can run their own analyses, and discuss caveats and validation strategies necessary when working with LLM-generated results. We contribute:

- A demonstration of unstructured data processing applied to avalanche accident and incident reports from the US and Canada (Section 3).
- A discussion of challenges including hallucination, prompt sensitivity, and reproducibility, with mitigation strategies (Section 4).
- A guide to available tools for researchers to conduct their own analyses (Section 5).

2. BACKGROUND

This section reviews relevant concepts, ideas, and background on unstructured data processing. Figure 2 presents an overview of the workflow.

2.1. *Unstructured Data Processing*

LLMs can interpret semantics and operate over text documents to perform tasks like summarization, extraction, and classification using a natural language specification of the task and the desired output format. This makes it possible to treat any document collection as a query-able dataset where an analyst specifies the task, and the LLM executes their specification against each document.

There are several ways to apply LLMs to a document collection. The simplest is to use LLM-powered agents to retrieve and read reports autonomously and answer analytical questions. But they have high cost per token, and their execution paths are difficult to inspect and reproduce. In this paper, we leverage DocETL, an LLM-powered unstructured data processing framework that can analyze documents systematically (Shankar et al. 2025a). Here, the analyst (or, an LLM-based coding agent on their behalf) authors a pipeline of operators that apply LLMs to process data — making pipelines amenable to cost and accuracy driven optimizations, while also making it possible for analysts to inspect and debug intermediate outputs.

Inputs to unstructured data processing pipelines are documents represented as collections of key-value pairs holding both structured fields (e.g., `date`, `location`, `group_size`) and unstructured narrative text. Each pipeline operator is defined by a *natural-language prompt,* and an *output schema* specifying the structure in which the LLM must respond. Operators can support varied input-output semantics, e.g., a `map` operator applies a prompt to each document to produce exactly one output per document; a `cluster` operator identifies groups and assigns a label to each document; a `reduce` or `aggregate` processes multiple documents together (often grouped by an attribute) to produce a

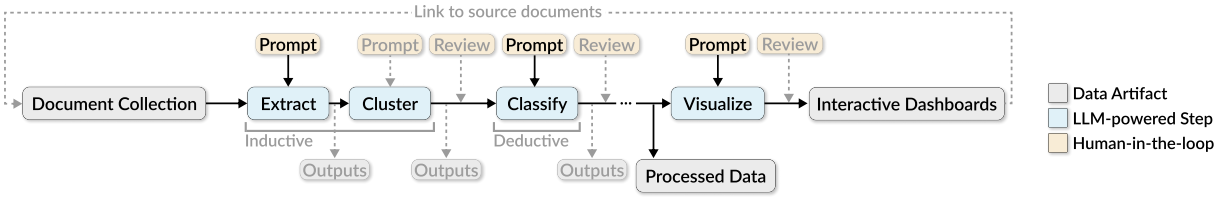


Figure 2: **Overview of the unstructured data processing workflow.** A document collection is processed through a pipeline of LLM-powered operators. Analysts optionally review intermediate outputs and refine operators. Interactive visualizations help inspect outputs with links to source documents for validation.

single output per group; `filter` keeps documents that meet a specified condition; while a `resolve` canonicalizes different names describing the same entity (e.g., both *"transceivers"* and *"beacons"* refer to the same technology). Operators can be chained together to form pipelines that incrementally build on prior outputs (Figure 2).

2.2. Deductive and Inductive Analysis

When working with a document collection, analysts may approach it with specific analytical questions and a pre-defined classification taxonomy, or with an open-ended intent to discover patterns that emerge from the data. Unstructured data processing pipelines can support both modes of analysis.

In a *deductive* (top-down) analysis, the task and output categories are known upfront. For example, an analyst might classify each accident report according to which of McCammon's heuristic traps are evident in the narrative, with the set of categories specified in advance. In an *inductive* (bottom-up) analysis, the analyst does not prescribe categories upfront. For instance, an analyst may be interested in observing common plan deviations to build on their analysis in Figure 1. An inductive classification can begin with extracting open-ended insights from each document, then clustering them into emergent categories. An analyst can then review outputs to merge or split categories, before feeding these refinements into a subsequent deductive classification step. In practice, most analyses combine deductive and inductive steps, as we illustrate in Section 3.

2.3. Interactive Visualizations and Dashboards

Analysts still need to make sense of pipeline outputs. Traditional data tooling (e.g., spreadsheets, Tableau, Python `pandas`, `R`, `MATLAB`) are primarily designed for structured data and are poorly suited to text-heavy inputs and outputs (e.g., free-text summaries, extracted quotes). Additionally, LLM-generated outputs may contain inaccuracies (discussed in Section 4)—making it crucial for analysts to validate, verify, and trace findings back to the source documents.

A practical solution is to also use LLMs downstream to generate purpose-built interactive dashboards and visualizations (Y. Cao et al. 2025), tailored to each analysis' sensemaking and validation requirements. We present a gallery of such interactive dashboards that demonstrate interactive drill-down charts to support comprehension, and link each output to its source to support provenance in Section 3 and Figure 3.

3. ASKING NEW QUESTIONS OF AVALANCHE REPORTS

To identify where unstructured data processing could be useful for avalanche accident report analysis, we reached out to forecasters and analysts at the Colorado Avalanche Information Center (CAIC) and Avalanche Canada. We gathered analytical questions they have had of their document collections which are challenging to answer without extensive human effort. These questions fell into three broad analysis types:

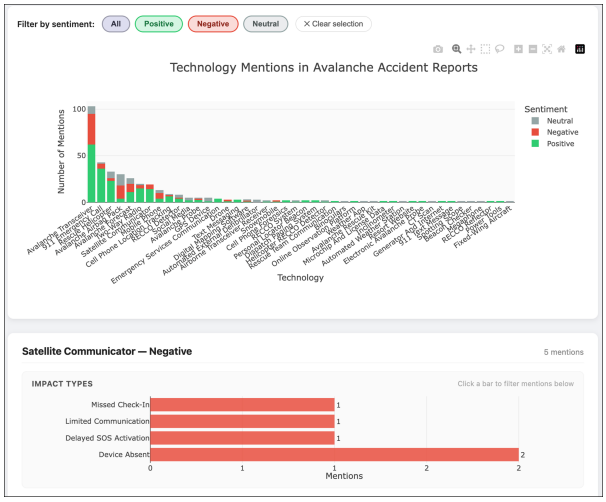
Extraction. Deriving new structured fields from narrative text—i.e., turning free-text descriptions into countable data for statistical analysis downstream. For example, classifying heuristic traps evident in each report.

Cross-document comparison. Applying the same analysis across two or more document collections grouped by an attribute (e.g., report type, source, time range, activity, region), and comparing outcomes. For example, comparing how adherence to plans differs between fatal and non-fatal reports, whether topic coverage is different in staff-authored and public-authored reports or whether solo-skier incidents describe different decision-making patterns than group ski tours.

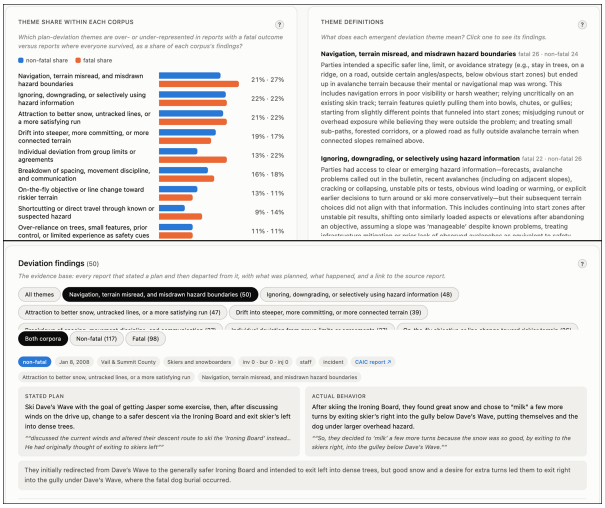
Narrative enrichment. Combining existing structured data fields with information extracted from narrative text to answer questions that

Table 1: **Overview of analyses.** Each analysis demonstrates a representative use of unstructured data processing for extraction, cross-document comparison, and narrative enrichment.

Analysis	Type	Description	Corpus used
Technology Use	Extraction	Extracting technology mentions from narratives, classifying sentiment, and identifying success and failure themes.	US fatal; Avalanche Canada
Plan Deviations	Comparison	Identifying deviations from stated plans and comparing emergent themes across fatal and non-fatal outcomes.	US fatal; CAIC non-fatal
Rescue Timelines	Enrichment	Reconstructing timelines using available timestamp fields and narrative description of rescue efforts.	CAIC non-fatal

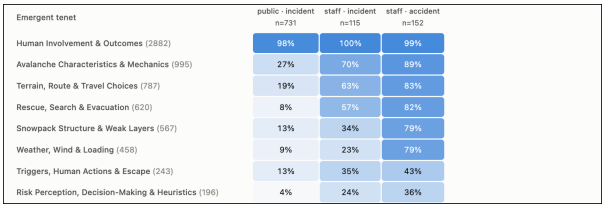


(a) **Technology Use.** Mentions of technology use, colored by sentiment. Clicking a bar segment shows themes and source reports; here, negative satellite communicator mentions.

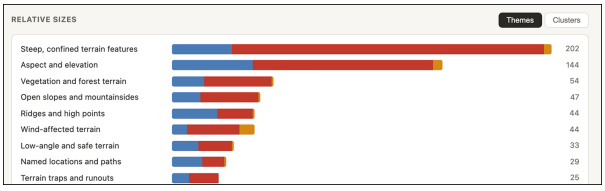


(b) **Plan Deviations.** Emergent deviation themes and their share for each corpus. Selecting a theme filters findings, each summarizing the stated plan against actual behavior.

(c) **Rescue Timelines.** Reconstructed timelines from narratives and available timestamps with aggregated statistics. Rescue efforts are represented as dots and link to source reports.



(d) **Tenet Coverage.** Comparing the prevalence of emergent tenets across staff- and public-authored reports. Clicking on a cell leads to source reports and excerpts corresponding to the selected tenet.



(e) **Terrain Descriptions.** Phrases used by skiers to describe terrain, clustered by features. Expanding a cluster reveals verbatim phrases with their relation to the incident.

Figure 3: Visualizing analysis outputs with LLM-generated dashboards.

neither can answer alone. For example, cross-referencing timestamps to re-construct rescue timelines, gathering supporting or contradicting evidence of avalanche risk/problem type or terrain details from first-hand narrations.

We demonstrate three representative pipelines across these analysis types. An overview is presented in Table 1 and Figure 3.

Data and Tooling

We analyzed three document collections:

- **US fatal accident reports** provided by CAIC,

comprising 419 reports written by investigators, covering fatalities across the United States from October 1994 to April 2026;

- **CAIC non-fatal incident reports**, comprising 1,001 reports (973 from Colorado) with narratives submitted by the public and CAIC investigators from March 1997 to May 2026; and
- **Avalanche Canada incidents**, comprising 577 fatal incident records across Canada from 1782 to April 2026, of which 450 include bilingual (English and French) narratives.

We used DocETL (Shankar et al. 2025a), an open-source unstructured data processing system developed at UC Berkeley, to build and run the processing pipelines described below, though our analyses do not depend on this toolset — they can also be carried out with other commercially available tooling (discussed in Section 5). Our analysis outputs and interactive dashboards are available at `snowtext.org`.

3.1. Technology Use

Accident narratives frequently mention technologies involved in rescues (e.g., transceivers, satellite communicators, airbags). However, the role each technology played is buried in prose and varies widely across reports. This pipeline extracts technology mentions, classifies the sentiment associated with each (positive, negative, or neutral), and groups them by success and failure themes. The insights from such an analysis could be used to make backcountry travelers aware of the limitations of their technologies. Because an exhaustive inventory of relevant technologies is hard to specify upfront, an initial open-ended `map` operator extracts technology mentions. A `resolve` step canonicalizes technology names so that, e.g., *"beacon"* and *"transceiver"* map to the same name. The clustering step discovers common success and failure themes inductively rather than requiring the analyst to anticipate them.

Satellite communication devices were positively mentioned when they were successfully used to notify emergency services about an accident. However, a number of reports also pointed out suboptimal use or functionality limitations, e.g., when delayed activation of the SOS feature led to delayed response, or the limited text communication bandwidth led to confusion about the response.

In seven reviewed accidents, location tracking of a cell phone of a group member was mentioned as a positive factor helping establish a search area. However, in at least one case the phone location data was inaccurate and misdirected the search.

Extract technology use — INDUCTIVE MAP
"Extract names of technologies mentioned in this report"

↓ *Technologies mentioned per report*

Classify sentiment of technology use — DEDUCTIVE MAP
"What was the sentiment associated with the use of this technology during rescue?"

↓ *Technologies with associated sentiments per report*

Resolve technology names — RESOLVE
"Give a canonical name to all duplicate technology names"

↓ *Canonicalized technologies with sentiments per report*

Identify success and failure themes — INDUCTIVE CLUSTER
"Given the technology and its sentiment for this group of reports, cluster them into reasons."

↓ *Canonical technologies, sentiments, and cluster themes*

Visualize outputs — VISUALIZE
"Generate an interactive stacked bar chart [...] clicking into a bar should reveal themes and underlying reports"

3.2. Plan Deviations

Accident and incident narratives may reveal deviations between stated plans and actual behavior. This pipeline runs on both the US fatal and CAIC non-fatal corpora to identify plan deviations and enable comparison across outcome severity. An initial `filter` step helps omit reports that do not indicate a plan deviation as unnecessarily processing them through downstream operators would add to cost. Filtered reports are then summarized and clustered into emergent deviation themes (e.g., *"Attraction to better snow," "Individual deviation from group limits or agreements"*).

Filter reports with plan deviations — FILTER
"Filter reports describing a stated plan or intention and describing actual behavior that differs from that plan"

↓ *Filtered reports*

Summarize deviation — INDUCTIVE MAP
"Summarize deviations from original plans or stated intents in this report"

↓ *Deviation summary per report*

Identify deviation themes — INDUCTIVE CLUSTER
"Cluster all summaries of plan deviations into themes"

↓ *Clustered deviation summaries*

Visualize outputs — VISUALIZE
"Generate a visualization comparing the share of deviation themes between fatal and non-fatal outcomes [...] present excerpts from stated plans and actual behavior with the deviation summary."

The dashboard (Figure 3b) compares the share of each emergent deviation theme across fatal and non-fatal outcome severity. Misread terrain, misdrawn hazard boundaries, and selective use of hazard information emerge as the most common deviations, while individual deviations from group limits and agreements shows the largest difference in share between non-fatal and fatal outcomes. Selecting a theme displays each report's stated plan, actual behavior, and deviation summary.

3.3. Rescue Timelines

Understanding how long different phases of rescue take requires combining information from structured fields and narrative text. The CAIC non-fatal corpus includes timestamp fields (`reported_at`, `dispatched_at`, `arrived_on_scene_at`) alongside prose descriptions of rescue efforts. However, timestamps are often sparse, and narratives lack temporal precision. This pipeline extracts an ordered sequence of rescue efforts from each narrative, anchors them to relative timestamps using structured and narrative references, and classifies the type of rescuer for each event.

Extract rescue sequence — INDUCTIVE MAP
"For each report, extract the sequence of rescue efforts post-avalanche as a list"

↓ *List of rescue efforts per report*

Reconstruct timeline — INDUCTIVE MAP
"Treat avalanche occurrence as T0. Use the report and available timestamps (reported_at, dispatched_at, or arrived_on_scene_at) to map each rescue effort to relative time of occurrence."

↓ *Relative time per rescue effort occurrence*

Identify rescuers — DEDUCTIVE MAP
"For each rescue effort in this report, classify the rescuer as: self, companion, bystander, organized rescue, [...]."

↓ *Relative time and rescuer per rescue effort occurrence*

Visualize outputs — VISUALIZE
"Visualize each rescue timeline on a asinh scale [...] Present stats on time to rescue per rescuer"

The dashboard (Figure 3c) plots individual rescue efforts within an incident as dots on a timeline, and also calculates statistics for different phases (burial, extrication, evacuation) by rescuer type. Although it is widely known that backcountry rescues can take many hours, the dashboard can reveal more detail about distinct rescue stages. Inspecting timelines also revealed four instances where structured timestamps were out of order (e.g., rescuers recorded as arriving before being dispatched), suggesting that such analyses can surface potential data-entry errors.

3.4. Additional Analyses

This approach can be extended to perform additional such analyses. For instance, we performed a tenet coverage analysis that asks what tenets are covered by staff- versus public-authored reports (Figure 3d); and a terrain description analysis that examines how skiers describe terrain in their own words and maps the relation of each feature to its involvement in the avalanche incident (Figure 3e).

4. CHALLENGES AND FUTURE WORK

Building unstructured data processing pipelines with LLMs introduces challenges at multiple stages of the workflow, including specifying analyses and validating outputs amid LLM hallucinations and inaccuracies. We discuss strategies to tackle these challenges and point to open directions for future work in unstructured data processing.

4.1. Authoring Pipelines

A central difficulty is writing prompts that reliably produce desired outputs across an entire document collection. First, analysts must translate their intent into a concrete prompt. For instance, an analyst asking *"what technology problems arise in rescues?"* must decide if text messaging and GPS count as technologies different from cell phones. Such decisions are hard to make upfront and typically require inspecting outputs from initial attempts. Second, even a well-crafted prompt may not generalize across all reports. Accident reports vary in narrative style, length, and detail—prompts tuned on limited examples may produce inconsistent outputs or miss edge-cases.

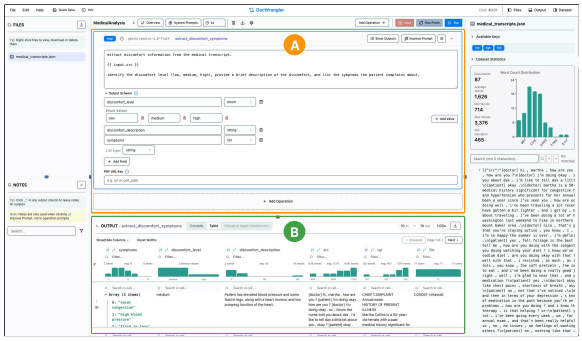


Figure 4: The DocWrangler interface, showing (A) a pipeline constructor and (B) a data inspector.

Interactive systems like DocWrangler (Shankar et al. 2025b) (Figure 4) help address both problems by letting analysts work with document samples, write pipelines, visually inspect outputs, and iteratively refine prompts. Analysts can annotate specific outputs in-situ (e.g., *"this should not count*

as a plan deviation"), and the system incorporates these notes to suggest improved prompts. When an operation attempts too much at once, DocWrangler detects accuracy issues and suggests decomposition into simpler operators.

4.2. Output Validation and Provenance

LLMs can hallucinate, producing plausible outputs that are not grounded in the source text. We encountered cases where the LLM inferred plan deviations from ambiguous phrasing that a domain expert would read differently, or attributed technology mentions to the wrong phase of an incident.

Interactive dashboards linking LLM outputs back to the source report and specific text passages are one way to mitigate risks by supporting provenance and expediting review. For instance, letting a reviewer click through individual technology mentions to verify whether the LLM correctly classified their role. But this does not eliminate the need for careful, domain-informed review. We also encountered cases where visualization code introduced its own errors, such as incorrect temporal arithmetic, independent of pipeline accuracy, underscoring that systematic validation of each workflow stage remains an open problem.

4.3. Reflexive Interpretation of Outputs

Beyond technical limitations, a more fundamental concern is that unstructured data processing turns qualitative narratives into structured, countable outputs. Narratives inherently reflect the biases of their authors and variance in reporting processes, and applying uniform automated treatment can obscure these differences. The three corpora we use illustrate this: CAIC fatal reports are written by investigators who frequently visit accident sites; non-fatal reports are primarily submitted by people possibly involved in a traumatic incident; and Avalanche Canada reports are written retrospectively with data from a coroner's inquiry. In addition to differences in provenance, people involved in traumatic events may not have accurate recollection of what happened. Aggregating findings across such sources can suggest patterns that are reflective of reporting practices as much as underlying phenomena. Thus, pipeline outputs are best treated as a starting point for analysis, and interpreting them calls for the same reflexivity expected in qualitative research.

4.4. Future Directions

Based on our experience and feedback from professionals at CAIC and Avalanche Canada, we identify two future directions.

In-dashboard interactive review. A recurring request was the ability to correct outputs within the dashboard, and having those corrections feed back into the processing pipeline. A natural extension would let experts relabel outputs, merge categories, or mark hallucinations directly in the dashboard, with corrections compiling into prompt or pipeline improvements for future analyses.

Supporting exploratory data analysis. Unstructured data processing makes it feasible to ask many questions in rapid succession, giving rise to a new challenge: understanding which questions a document collection can and cannot answer. Future tooling could expedite this exploration by proactively surfacing patterns, themes, anomalies, and analytical directions stemming from a document collection, helping analysts quickly identify questions their data can productively answer.

5. AVAILABLE TOOLING FOR UNSTRUCTURED DATA PROCESSING

Several tools can support unstructured data processing. LLM coding agents (e.g., Cursor, Claude Code, Codex) can be prompted to generate pipelines, and are a practical starting point. Major database systems including Snowflake Cortex (Liskowski et al. 2026) and Google BigQuery (Google Cloud 2024) now offer built-in semantic operators (e.g., filter, classify, aggregate) that can be applied to text columns in SQL. These platforms lower the barrier for organizations already using cloud data infrastructure, though they currently offer less control and human oversight over than the open-source systems below.

Pipeline authoring systems. For the analyses presented in this paper, open-source unstructured data processing systems like DocETL (Shankar et al. 2025a), LOTUS (Patel et al. 2025), and Palimpzest (Liu et al. 2025) provide composable semantic operators that can be chained into pipelines over document collections. DocWrangler (Shankar et al. 2025b) adds an interactive interface on top of DocETL to help users iteratively inspect and refine their pipelines.

Inductive operators. Several research tools implement inductive operators that can be used as independent, composable steps in unstructured data processing pipelines. LLooM (Lam et al. 2024) is one such approach, that implements LLM-powered inductive concept coding to discover themes from text collections bottom-up. Several tools also support tasks like classification, cluster-

ing, ranking, and comparative scoring that require LLMs to have a holistic understanding of documents to derive outputs bottom-up (Chopra et al. 2026; Sun et al. 2026).

6. CONCLUSION

We demonstrated how LLM-powered unstructured data processing can extract insights from avalanche reports, enabling analyses that would otherwise require extensive manual effort. Questions posed by forecasters and analysts fell into three categories—extraction, cross-collection comparison, and narrative enrichment. We introduce open-source tooling to run such analyses as pipelines composed of semantic operators. We discuss challenges with authoring pipelines and validating LLM outputs, while presenting potential mitigation strategies like the use of interactive dashboards with in-situ provenance for expert review. We invite the avalanche community to explore the analyses presented in this paper, and to consider applications of unstructured data processing for their own document collections.

ACKNOWLEDGEMENTS

We thank Mike Cooperstein, Ethan Greene, Spencer Logan, Simon Horton, and Grant Helgeson for valuable discussions and feedback on our analyses, and for providing access to CAIC and Avalanche Canada accident and incident reports.

REFERENCES

Armstrong, B. R. (1980). “Avalanche Burials in the United States: 1967–1980”. In: Proceedings of the 1980 International Snow Science Workshop. Vancouver, BC, Canada, pp. 215–230.

Atkins, D. (2000). “Human Factors in Avalanche Accidents”. In: Proceedings of the 2000 International Snow Science Workshop. Big Sky, Montana, USA, pp. 46–51.

Cao, Y., Jiang, P., and Xia, H. (2025). “Generative and Malleable User Interfaces with Generative and Evolving Task-Driven Data Model”. In: Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems. CHI ’25. ACM. ISBN: 9798400713941. https://doi.org/10.1145/3706598.3713285.

Chopra, B., Chen, M., Dang, R., Park, C., Shankar, S., Zeighami, S., Hartmann, B., and Parameswaran, A. G. (2026). “Who’s Keeping Score? Interactive Steering of LLM-Powered Scoring with Attune”. In: Proceedings of the 39th Annual ACM Symposium on User Interface Software and Technology (UIST ’26). ACM. https://doi.org/10.1145/3830398.3830567.

Gallagher, J. E. (1967). The Snowy Torrents: Avalanche Accidents in the United States, 1910–1966. Washington, DC: U.S. Department of Agriculture, Forest Service.

Google Cloud (2024). BigQuery ML: Generative AI Functions. `https://cloud.google.com/bigquery/docs/generative-ai-overview`. Accessed: 2026-08-19.

Hayes, A. S. (2025). “Conversing” With Qualitative Data: Enhancing Qualitative Research Through Large Language Models (LLMs). In: International Journal of Qualitative Methods 24. https://doi.org/10.1177/16094069251322346.

Lam, M. S., Teoh, J., Landay, J. A., Heer, J., and Bernstein, M. S. (2024). “Concept Induction: Analyzing Unstructured Text with High-Level Concepts Using LLooM”. In: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI ’24). ACM. https://doi.org/10.1145/3613904.3642830.

Liskowski, P., Han, B., Aggarwal, P., Chen, B., Jiang, B., Jindal, N., Li, Z., Lin, A., Schmaus, K., Tayade, J., et al. (2026). “Cortex AISQL: A Production SQL Engine for Unstructured Data”. In: Companion of the International Conference on Management of Data, pp. 400–412.

Liu, C., Russo, M., Cafarella, M. J., Cao, L., Chen, P. B., Chen, Z., Franklin, M. J., Kraska, T., Madden, S., Shahout, R., et al. (2025). “Palimpzest: Optimizing AI-Powered Analytics with Declarative Query Processing.” In: CIDR. https://doi.org/10.48550/arXiv.2405.14696.

Logan, S. and Greene, E. (2014). “The Distribution of Fatalities by Avalanche Problem in Colorado, USA, 1998–99 to 2012–13”. In: Proceedings of the 2014 International Snow Science Workshop. Banff, Alberta, Canada.

McCammon, I. (2002). “Evidence of Heuristic Traps in Recreational Avalanche Accidents”. In: Proceedings of the 2002 International Snow Science Workshop. Penticton, British Columbia, Canada, pp. 244–251.

McCammon, I. (2004). Heuristic Traps in Recreational Avalanche Accidents: Evidence and Implications. In: Avalanche News 68, pp. 42–50.

Mellon, J., Bailey, J., Scott, R., Breckwoldt, J., Miori, M., and Schmedeman, P. (2024). Do AIs know what the most important issue is? Using language models to code open-text social survey responses at scale. In: Research & Politics 11.1. https://doi.org/10.1177/20531680241231468.

Patel, L., Jha, S., Pan, M., Gupta, H., Asawa, P., Guestrin, C., and Zaharia, M. (2025). Semantic Operators and Their Optimization: Enabling LLM-Based Data Processing with Accuracy Guarantees in LOTUS. In: Proceedings of the VLDB Endowment 18.11, pp. 4171–4184. https://doi.org/10.14778/3749646.3749685.

Shankar, S., Chambers, T., Shah, T., Parameswaran, A. G., and Wu, E. (2025a). DocETL: Agentic Query Rewriting and Evaluation for Complex Document Processing. In: Proceedings of the VLDB Endowment 18.9, pp. 3035–3048. https://doi.org/10.14778/3746405.3746426.

Shankar, S., Chopra, B., Hasan, M., Lee, S., Hartmann, B., Hellerstein, J. M., Parameswaran, A. G., and Wu, E. (2025b). “Steering Semantic Data Processing With DocWrangler”. In: Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST ’25). ACM. https://doi.org/10.1145/3746059.3747625.

Sun, Y., Zeighami, S., Chopra, B., Shankar, S., and Parameswaran, A. G. (2026). Semantic Data Processing with Holistic Data Understanding.

Techel, F., Jarry, F., Kronthaler, G., Mitterer, S., Nairz, P., Pavšek, M., Valt, M., and Darms, G. (2016). Avalanche Fatalities in the European Alps: Long-Term Trends and Statistics. In: Geographica Helvetica 71.2, pp. 147–159. https://doi.org/10.5194/gh-71-147-2016.

Than, N., Fan, L., Law, T., Nelson, L. K., and McCall, L. (2025). Updating “The Future of Coding”: Qualitative Coding with Generative Large Language Models. In: Sociological Methods & Research 54.3, pp. 849–888. https://doi.org/10.1177/00491241251339188.