Automatic Histograms: Leveraging Language Models for Text Dataset Exploration

Emily Reif*
ereif@google.com
Google Research
Seattle, WA, USA

Crystal Qian*
cjqian@google.com
Google Research
New York City, NY, USA

James Wexler
jwexler@google.com
Google Research
Cambridge, MA, USA

Minsuk Kahng
kahng@google.com
Google Research
Atlanta, GA, USA

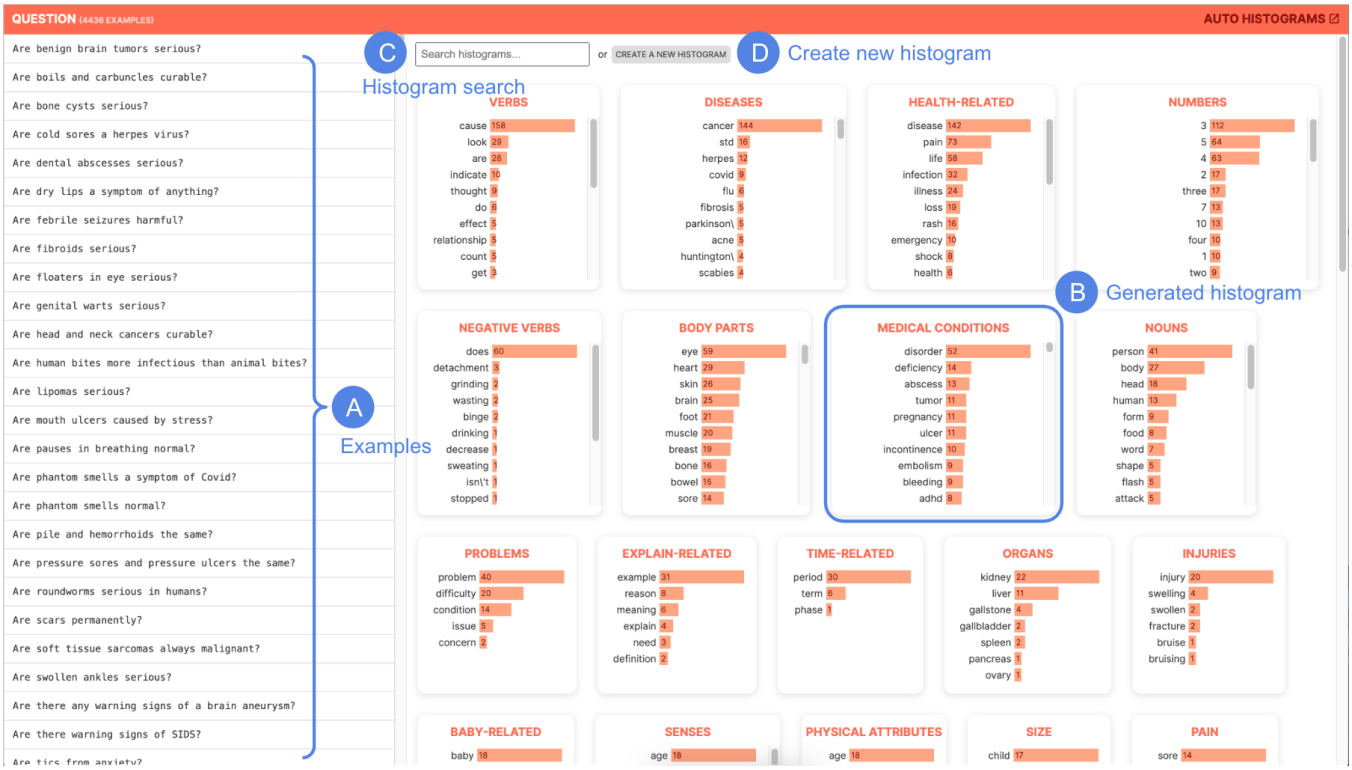

Figure 1: ***AutoHistograms: A tool for making sense of unstructured text datasets.*** **(A) Dataset examples are shown on the left hand side. (B) Dataset-specific distributions of entities in the dataset are generated in a pre-processing step, and visualized with bar charts. (C) Distributions can be searched with exact string matching or semantic search. (D) New distributions can be generated in real time for in-the-loop dataset exploration.**

ABSTRACT

Making sense of unstructured text datasets is perennially difficult, yet increasingly relevant with Large Language Models. Data practitioners often rely on dataset summaries, especially distributions of various derived features. Some features, like toxicity or topics, are relevant to many datasets, but many interesting features are domain specific: instruments and genres for a music dataset, or diseases and symptoms for a medical dataset. Accordingly, data practitioners often run custom analyses for each dataset, which is cumbersome and difficult, or use unsupervised methods. We present AutoHistograms, a visualization tool leveraging LLMs. AutoHistograms automatically identifies relevant entity-based features, visualizes them, and allows the user to interactively query the dataset for new categories of entities. In a user study with (n=10) data practitioners, we observe that participants were able to quickly onboard to AutoHistograms, use the tool to identify actionable insights, and conceptualize a broad range of applicable use cases. Together, this tool and user study contribute to the growing field of LLM-assisted sensemaking tools.

*Both authors contributed equally to this research.

CHI '24, May 11–16, 2024, Honolulu, HI

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/3613905.3650798

CCS CONCEPTS

• **Computing methodologies** → **Natural language generation**; **Topic modeling**; • **Human-centered computing** → **Information visualization**.

ACM Reference Format:
Emily Reif, Crystal Qian, James Wexler, and Minsuk Kahng. 2024. Automatic Histograms: Leveraging Language Models for Text Dataset Exploration. In *CHI '24: ACM conference on Human Factors in Computing Systems, May 11–16, 2024, Honolulu, HI*. ACM, New York, NY, USA, 9 pages. https://doi.org/3613905.3650798

1 INTRODUCTION

Making sense of unstructured text datasets is an increasingly important, unsolved challenge. There are many high-stakes use cases where it is essential [30], especially with the rise of large language models (LLMs). These include curating the pre-training and finetuning datasets of LLMs, and creating evaluation benchmark datasets for areas such as safety, factuality, or other desired behaviors. Ideally, there would be quantitative methods to determine if a dataset is high quality. However, given that many of these LLM tasks are open-ended (e.g., creative writing, summarizing, or question answering), standard accuracy metrics can be inappropriate or insufficient [19], as there is often no ground truth at all. To determine if a dataset is of sufficient quality, the data practitioner [28] must first define what quality means in the context of the dataset. To do this, they must qualitatively understand the dataset itself.

As it is usually impossible to read every example in a dataset [33], many of the analyses for understanding unstructured text datasets center around calculating distributions and diversity [24] along specific derived features of the text. The field is converging on which features are applicable across many datasets (e.g., toxicity, topics, or protected groups) and formalizing them into frameworks [14, 18, 27]. There are also pipelines [5, 16] and visualization tools [1–4] to annotate and explore these broadly-applicable features. These NLP-based methods and tools have been empirically shown to improve sensemaking and qualitative data understanding practices [20, 22, 25].

However, these supervised methods are not enough on their own; data practitioners also care about dataset-specific analyses or features. Unsupervised methods for text dataset analysis start to fill this gap. Topic modeling [9] and neural topic modeling [21, 32], define topics in a dataset, and determine which apply to a given example. Examples can also be embedded with a pretrained model [29], and then clustered or visualized [7, 10, 12, 23]. These unsupervised methods have drawbacks as well, though. It is often difficult to understand what a cluster or topic represents, potentially leading to issues in downstream tasks. For example, filtering out an unsupervised topic can have unintended consequences if it is unclear exactly what the topic contains. It can also be hard to communicate the meaning of these clusters to other stakeholders.

To bridge this gap, we present AutoHistograms.[1] AutoHistograms is a semi-supervised method and visualization tool that automatically extracts semantically-meaningful entity-based features from raw unstructured text. It then displays interactive visualizations of their distributions in the form of bar charts, and allows for real-time calculation of distributions of features queried by the user. AutoHistograms is open source[2], and leverages LLMs' generative abilities and rich embedding spaces to cluster domain-specific features. Given a dataset, it automatically calculates the distribution of specific categories of entities relevant to that dataset. For example, if the dataset contains mentions of "covid 19", "the flu", and "SARs", AutoHistograms groups these terms and produces a distributions of "infectious diseases" in the dataset (see Section 4.1). Users can also easily create new distributions in real time for in-the-loop hypothesis testing. For example, they can query a dataset with the natural language description of "body parts" without having to define a preset list of all possible body parts to run the analysis (see Section 4.3).

We also present a user study with 10 data practitioners and data tool creators to evaluate AutoHistograms. Participants ramped up quickly and were able to perform actions defined within our key user journeys with minimal assistance. They were able to flag contextually relevant features of the dataset using the tool, and identified opportunities to apply AutoHistograms in a range of other use cases, including verifying safety, detecting outliers, debiasing example selection, and identifying mode collapse in synthetic data.

2 USER CHALLENGES

The following user challenges are based on previous informal conversations with data practitioners at Google where the authors make tools for evaluating training, benchmark, or synthetically generated data. Prior literature has already identified a range of common user needs around navigating a dataset [6]. We focus on the challenges of users who finetune and evaluate LLMs, and thus need to develop a qualitative understanding of unstructured text datasets.

- **C1: Summarize the dataset with relevant distributions.** As mentioned in Section 1, one common practice is to look at distributions of derived text features. As annotating and analyzing dataset-agnostic features is already well-supported, we focus on categorical features that are *specific to a given dataset*. For example, someone curating a dataset for a music recommendation system might care about the distribution of genres, instruments, or artists, but someone making a more targeted responsible AI benchmark dataset might care about the distribution of specific religions, genders, or races. There are a few specific steps in this process:
  a) **Determining relevant features.** In a novel dataset, it is not always immediately clear what features will provide interesting insights in the data.
  b) **Annotating identified features.** It can be difficult to annotate data with these features, especially at scale.
  c) **Displaying the feature distributions.** After each example is annotated with the feature, it is necessary to have some form of visualization or summary of the feature across the dataset.
- **C2: Find pathological distributions.** While this shares many of the low-level implementation challenges as **C1**,

[1] While the term "histogram" usually refers to the visualization of a numeric value, we use it here for categorical values as well.

[2] https://github.com/PAIR-code/auto-histograms

finding imbalances in the derived features is often described as a separate high-level goal.

- **C3: Find surprising slices of data.** Complementary to the summary, users also need to find groups of examples that wouldn't necessarily be captured in the main summaries or by a quick scan. For example, a medical dataset might have a group of examples suggesting fringe medical advice. There might be so few examples that these would not be found by either a quick overview of the dataset, or in the main summaries.
- **C4: Onboard quickly.** While not directly related to dataset understanding, an essential need that is often overlooked is being able to actually use these tools without too much startup cost. Ideally, these tools would be automatically integrated into standard workflows.

3 DESIGN GOALS

To address these unmet needs, we designed our tool with the following goals in mind.

- **G1: Automatically show salient features.** To address **C1** and **C2**, AutoHistograms must determine what features are relevant to a dataset. To address **C4**, this must be unsupervised; the user should not have to predefine all the features of interest.
- **G2: Let users quickly iterate on these features.** However, this feature selection will not always be perfect. Also in support of **C1**, users should be able to add to the automatically-generated distributions by creating new ones in real time. This freeform exploration also supports **C2**.
- **G3: Visualize feature distributions** To support **C1-3**, the tool should display the feature distributions in an easily digestible format. For **C3**, specifically, it should let the user interactively dig into the specific examples that belong to a bucket of a given distribution.

Note that AutoHistograms specifically addresses the issue of finding dataset-specific distributions. We have integrated it into a general purpose dataset analysis and curation tool at Google, and we expect that it will generally be used in conjunction with supervised methods.

4 SYSTEM IMPLEMENTATION

AutoHistograms has three components:

(1) A pre-processing pipeline to calculate the distributions from the dataset.
(2) A visualization tool for viewing the generated distributions.
(3) A method for calculating new distributions interactively.

4.1 Calculating distributions

In this section, we describe the method for determining and calculating the relevant distributions for a dataset.

Extract entities The first step is to collect all entities across the dataset. We use NLTK [8] to select the nouns and numbers in the dataset. For performance reasons, we keep the most frequent k=2000 entities.

Cluster entities with embeddings We then find meaningful groups of entities to create the distributions. To do this, we calculate the embedding of each entity using the externally-available PaLM API [3], then cluster the entities in the embedding space using hierarchical clustering[4] with the *maxclust* criterion. It is desirable for a given entity (e.g., "email") to be present in multiple distributions (e.g.,"communications", "computer-related"), so we conduct multiple rounds of clustering on the dataset, varying the value of t from one to k = the total number of entities (t denotes the desired number of clusters provided as input to the hierarchical clustering algorithm). We reject clusters that contain less than three or more than 15 entities. The final set of clusters is the concatenation of these multiple clustering rounds.

Label distributions with LLMs We use the externally-available PaLM API[5] to label the groups of entities using a few shot prompt (see Appendix - A.1). We also filter out clusters that are classified low quality by the model.

4.2 Interactive exploration

The UI (Figure 1) allows the user to interactively explore the distributions and create new ones. The left side (Figure 1(A)) is a scrollable list of examples. The right side contains the automatically generated distributions (Figure 1(B)). When the user selects an entity in a distribution (Figure 4), the entity is highlighted, and the data table is filtered to only show examples that contain that entity. The user can also search for distributions by name, which will return exact or semantic similar matches (e.g., searching "diseases" also surfaces "illnesses".) See Figure 4. The UI is implemented using TypeScript and the LIT framework.[6]

4.3 New distributions in real time

If the user would like to explore a feature that was not automatically generated as part of the pipeline (e.g., find all the sexually transmitted diseases in the dataset), they can create a new distribution in real time with a human-in-the-loop process. This method leverages LLMs and embeddings to create a zero-shot classifier, using only one LLM inference call:

(1) User types query (e.g., "sexually transmitted diseases").
(2) The LLM is queried to "give me examples of <new feature name, e.g. sexually transmitted diseases>" (see Appendix - A.2). It returns some exemplar entities, which may or may not actually be in the dataset.
(3) Given these LLM-generated exemplars, we suggest semantically similar entities in the dataset by creating a Scikit-Learn[26] KNN classifier in the same embedding space from Section 4.3. To create the vector for the new query, we embed the LLM-generated exemplars using the same embedding model, and take their centroid. We then surface the entities in the dataset that have high embedding cosine similarity to this centroid, reusing the pre-computed embeddings in Section 4.3.

[3] https://ai.google/discover/palm2/ , text-gecko model
[4] https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html
[5] https://ai.google/discover/palm2/
[6] https://lit.dev

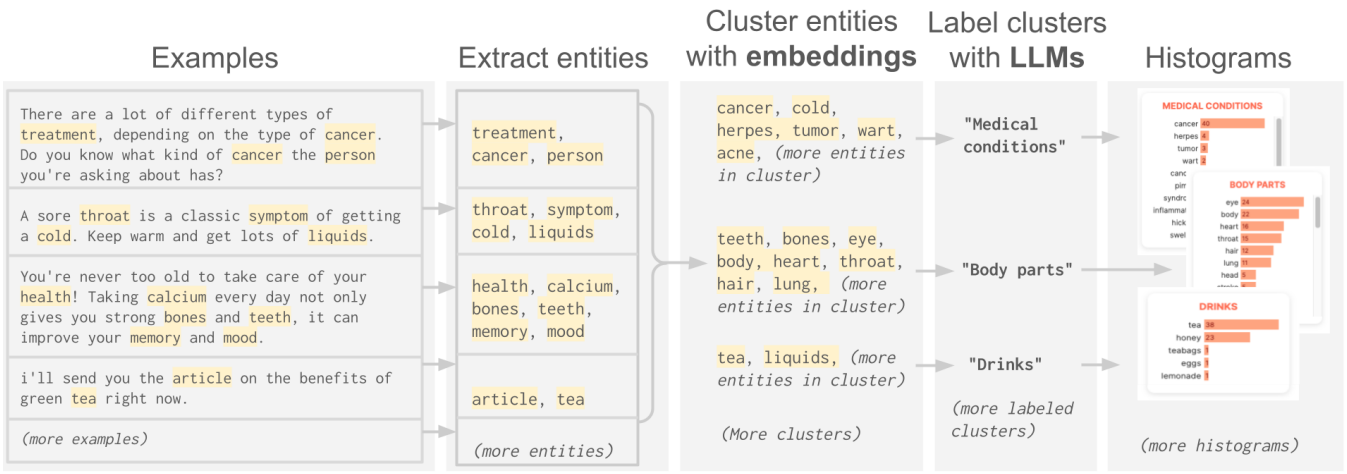

Figure 2: The processing steps for automatically creating distributions from a dataset

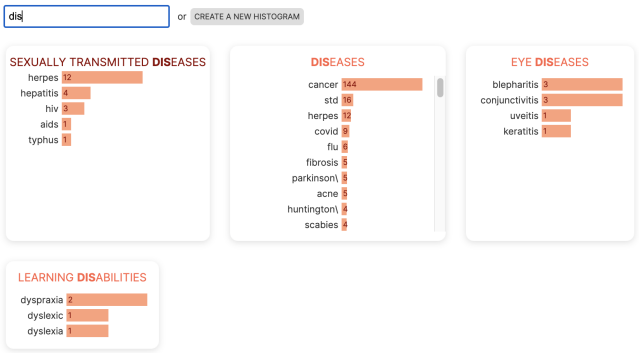

Figure 3: A search interface that supports exact or semantic search of categories.

(4) These semantically similar entities are then presented to the user, who selects entities to include in the distribution.
(5) Finally, we create the distribution of examples with each entity, and integrate it back into the UI.

5 TECHNICAL EVALUATION

In Section 6, we evaluate the tool end-to-end with a set of user studies. Here, we evaluate the components of the algorithm itself.

5.1 Desiderata of distribution calculation

What are the desired qualities of the unsupervised distributions calculated by our pipeline, and how can we measure them?

- **Distribution specificity**: Distributions should be specific to a given dataset. We measure this by calculating 1 - fraction of overlapping distribution labels between two different datasets.
- **Distribution accuracy**: Distributions should be relevant to a given dataset. There is no ground truth for this, so instead we compare against datasets with ground-truth topics as an approximation, calculating the percentage of ground-truth topics had a match in our computed topics. Two topics are considered a "match" if they have any lemmas in common, as calculated by NLTK.
- **Coverage**: The distributions should cover the full dataset. We measure this by calculating the percentage of examples that are contained in at least one distribution.
- **Cluster label accuracy**: A cluster of entities that define a distribution should be labeled correctly. We measure our labeling method with clusters from WordNet [17]. WordNet provides the list of hyponyms of a given word in its database (e.g., 'red' is a hyponyms of 'color'). We then attempt to label these hyponym lists using our labeling method. We report the success rate, where a labeling is a 'success' if our returned label is either an exact match of the word, or contains it (e.g., 'art' and 'art-related' would be a match). Note that WordNet contains a much more limited word list than what AutoHistograms supports, which is why we do not use it directly in our tool.

Baselines: We compare our method to two baselines, unsupervised **LDA** and **WordLists**. We use Scikit-Learn's [26] implementation of **LDA** [9], first removing NLTK's [8] stopwords, and leaving the number of components as the default (ten). On the other side of the spectrum, **WordLists** is a set of hard-coded word lists (religions, races, genders, professions, and bad words) that are used frequently by dataset practitioners for dataset analysis. These are not dataset specific. There are a few variations of these word lists, so we do a best-effort attempt at using the standard ones by aggregating those from [16], [13], and [15]. Specifically, the categories are "toxic words", "pronouns", "religions", "races" and "professions". For the full list of words, see Appendix - A.3.

Datasets We compare AutoHistograms and our baselines over three datasets from [31], which have ground-truth topic labels (**dbpedia**, **agnews**, and **nyt**).

Figure 4: When the bucket for an entity is selected, the data table is filtered to only examples that contain that entity.

	AutoHistograms	LDA	Wordlists
Distribution specificity	61.2% ± 0.6	100% ± 0.0	0% ± 0
Distribution accuracy	86.7% ±11.7	17.8% ± 18.1	12.7% ± 10.2
Coverage	98.4% ± 1.8	100% ± 0	25.7% ± 25.9
Cluster label accuracy	69.5% ± 6.5	n/a	100%± 0

Table 1: Automatic evaluation of AutoHistograms and baselines. AutoHistograms has high specificity and accuracy, while still maintaining high coverage of the dataset.

5.2 Results of technical evaluation

We find that AutoHistograms performs favorably compared to our baselines. The calculated distributions contain most of the ground truth topics (86.7%, on average) while covering almost the entire dataset (98.4% on average), and our labeling algorithm acheives 69.5% accuracy. **LDA** is, by construction, most specific to each dataset. However, looking at individual results from AutoHistograms, we find that the distributions that are repeated for multiple datasets are things like "locations", "names", etc, which do appear to actually be applicable to multiple datasets. The **Wordlists**, while having 100% label accuracy by construction, only end up cover a small percentage of the dataset (25.7%), and most of that is because of the "pronouns" distribution.

However, as we discussed before, all of these methods have strengths and weaknesses, and ideally would be used together to analyze different aspects of a dataset.

6 OBSERVATIONAL STUDY

Next, we conduct an observational study of users interacting with the Automatic Histograms tool, to validate the alignment of user behaviors with anticipated patterns. The structure of each individual, 30-minute, virtual user study is as follows:

- **Introduction** (5 min): Participant describes their background and use cases
- **Demo** (5 min): Moderator briefly demonstrates AutoHistograms on a sample dataset of musical terms, including the search and "create new" features.
- **Free-form exploration** (20 min): Participant follows a link to the tool and explores a dataset of sample chatbot responses to medical queries, sharing their screens with the moderator and thinking aloud. During this free-form exploration, participants took different angles of their choice, such as deciding whether to evaluate this dataset for safety concerns or cleaning the dataset for any apparent outliers to train a large language model.

6.1 Participants

We recruited 10 industry professionals at a large technology company (N=10) who have experience curating, analyzing, or using text-based datasets.[7] In our sample, none reported having physical limitations; four identified as female, six as male, and all are based across the US (Bay Area, New York, Atlanta, remote). Table 2 summaries their relevant experience. These participants included tool developers, model developers, and researchers, and had job titles of either "software engineer" (8) or "research scientist" (2). Most of the data they interact with are for the purpose of training and evaluating large language models. As part of the background interview, we asked participants what visualization tools they used to understand their data. For the most part, they are not currently using visualization techniques, but they want to (emphasizing **C1c**). Although they occasionally use tools like Jupyter or Colab, data exploration is usually performed by visually scanning examples in a .csv or spreadsheet. Most participants would like to use and explore visualization tools further, but do not due to the difficulty of easily creating them.

> *"I probably would [look at visualizations] if I took the time to build that. I tried to create a pie chart to show the distribution, but I just left it as a table. I was too lazy."* —P2

6.2 Observations

We conducted a thematic analysis [11] to analyze and code behaviors and commentary from the user sessions. Across the 10 user

[7] Note that Qian et al. [28] uses the same participant sample.

Participant	Tool Experience	Job Description
P1	No	Builds data pipelines for training conversational LLMs
P2	No	Generates synthetic data for adversarial testing of LLMs
P3	Yes	Builds tools for curating and annotating text-based datasets
P4	No	Develops text-to-image models using multimodal datasets
P5	Yes	Builds tools for curating and interpreting text-based datasets
P6	No	Conducts mix-methods research on data annotation agreement
P7	No	Uses LLMs to automatically rate benchmarking datasets for debugging model behaviors
P8	Yes	Conducts qualitative research on data annotation subjectivity
P9	No	Develops tools for improved data labeling and understanding
P10	No	Builds text-based datasets from webpage sources

Table 2: A summary of participants and relevant experience. "Tool experience" indicates that the user has viewed or interacted with previous iterations of the Automatic Histograms tool.

studies, we found evidence that the AutoHistograms addressed all four targeted user challenges.

C1, Identifying subject matter: Participants were not told the subject matter of this dataset. However, all participants correctly quickly identified the subject as *chatbot responses to medical queries*. They accomplished this by referencing the distribution panel and then validating in the data panel, rather than manually scrolling through the list of examples: 9/10 users initially focused on the distributions rather than the list of examples. Participants appeared to heavily rely on and interact with the distributions panel to synthesize their insights; they would refer to the list of examples primarily only to validate hypotheses and insights generated from looking at the distributions panel.

C2, Finding pathological distributions: Participants spent more time studying distribution with higher entropy, demonstrated by hovering, scrolling, and clicking on these long-tailed histograms. Six participants commented on the *diseases* bar chart in Figure 1, which had 144 instances of the token "cancer," 130 more instances than than the second-most frequent token.

> *"'I'm interested in surprises- for example, long tails. There's lots of cancer but not other medical conditions."* —P3

> *"Say I wanted to create a dataset that's balanced across diseases.. this tells me that it's [focused on] cancer. These titles [of distributions] tell me everything I need to know about the dataset."* —P3

C3, Identifying unexpected slices of data:

The distributions are sorted by the total count of number of occurrences. This can cause seemingly-arbitrary concepts to appear at the top of the interface; for example, in Figure 1, distributions about *verbs* and *numbers* appear next to *diseases* and *health*. The participants who were on the tool-building side appeared to be skeptical about the relevance of distributions for more ambiguous terms such as *things* and *ways*. However, the participants who performed more data analysis in their work reported liking that the distributions did not appear to be completely relevant. Adding a feature to specify sort order may help users to parse their data in ways to suit their needs. A common theme is that users wanted to be surprised by outliers; entropy was suggested three times as a sorting mechanism.

> *"It's neat that it's surfacing relevant tokens.. But not all histograms are useful."* —P5, a tool builder

> *"I like that there are seemingly less relevant suggestions of histograms (e.g. question words) because there can be surprising things. It was helpful, but I don't [typically] think that way."* —P6, a data scientist

> *"It's difficult to see where to look. You might look at 10 different directions and still nothing comes out until the 11th direction. We have potentially a hunch on what would be interesting to look at.. But we [are looking to be] surprised by what we see."* —P7

Participants were able to quickly identify and select interesting slices of data. Participants, not only those who worked on AI safety, wanted to ensure that the chatbot was not giving unsafe advice. Four participants typed the word "advice" into the search bar or created a new chart of "advice"-related terms to explore examples with this term. Using this workflow, participants were able to quickly flag potentially problematic examples of chatbot responses, such as "I'm not a doctor, but lemon and tea usually work for me."

> *"[Thinking about safety] is required.. especially in generative AI, there's strict review to make sure that your generated information is actually safe."* —P4

> *"This is just something borderline unsafe.. you're not supposed to give medical advice. For the bot to say that 'I would not want to give you medical advice, but... That is a safety violation."* —P8

> *"I'm hoping that [I'm not seeing] chatbot interactions related to health concerns because that would be against [company] policies."* —P5

C4: Onboarding and hypothesis-testing quickly: All participants were able to independently accomplish the above tasks with largely no intervention from the moderator. Participants were eager to actively interact with the tool; participants ubiquitously clicked on bars and expected the relevant data panels to appear on the left data panel. Rather than passively view the distributions panels, they actively scrolled, clicked on bars, typed in search queries, and interacted with the UI to address their dataset hypotheses.

> *"I immediately want to click this [bar] and see how many times it [appears]..."* —P7

6.3 Use cases

Participants identified the following use cases as opportunities to integrate this tool into their existing workflows:

Classification/tagging: Participants voiced that AutoHistograms could help them to understand the contents of a large dataset quickly and with less bias than their current method of manually reading a few select examples.

> *"If you had no prior information about the data... instead of reading all of the individual examples, you can read these [distribution lists]. Without this, I would have shuffled this data and then read it. I would have read a couple hundred before my eyes started bleeding."* —P9

Misclassification/identifying outliers: Using their current methods, data practitioners need to formulate hypotheses about existing outliers/bad data before finding them. By grouping tokens into buckets, AutoHistograms allowed users to quickly identify high-spread distributions such as cancer vs. other ailments.

> *"'We have these datasets that are supposedly of good quality. If you eyeball random examples, you can see that it's wrong, but you don't know how widespread that is in your dataset."* —P7

Safety: In terms of specific use cases, almost all participants identified *safety* as an area where AutoHistograms could have a positive impact. Particularly as generative models have become more pervasive, our participants stressed the importance of making sure that models are trained on safe data. AutoHistograms could help to identify correlations that appear in harmful queries (P2) and identify subsets of the finetuning data to rebalance such that toxicity scores are below a compliant threshold (P6). AutoHistograms could help to label generated content that violate safety standards (such as by giving medical or legal advice), and identify sensitive or adversarial topics (e.g. religion, politics) (P7).

Fairness: Fairness was another common use case: data can be rebalanced to ensure better representation amongst subgroups (P4) and models can be fine-tuned with evenly-distributed synthetic data if biases are discovered (P8).

Synthetic data: AutoHistograms could be used to identify mode collapse in synthetically generated data (P7).

7 LIMITATIONS AND CHALLENGES

Finally, we discuss the most common user feedback on the tool's limitations and challenges.

1. Participants also want to explore numerical metadata: As discussed in the introduction, there are general (non-dataset-specific) features that are useful for analysis. Three participants (P1, P4, P7) said they might look at numerical features such as text length, number of examples, token counts, and summary statistics. We have since integrated AutoHistograms into a general purpose data analysis tool at Google, which supports these features.

2: Demand for intersectional slicing: Balancing skews and uneven distributions of data appeared to be a key use case for data practitioners. Many participants wanted more fine-grained intersectional exploration in the tool, and asked for advanced searching (such as "AND" and "OR") clauses to support this need:

> *"[We're interested in] identity terms and formality of language. What kinds of topics come up [for different subgroups of annotators]? What is or is not being represented?"* —P6, on social-cultural context for annotator agreement

> *"The way that speaking is gendered is very subtle.. so [I'd want to categorize by] by word type, verbs."* —P9, on representation in conversational datasets

Participants also listed integration, speed, and reliability as key factors that would help them adopt AutoHistograms.

3: Drawbacks of using LLMs: While not explicitly noted by our participants, one drawback of this method is the lack of interpretability due to the use of LLMs, which are inherently black boxes. While more explicit methods of entity categorization (e.g., a knowledge graph) would have higher fidelity, we chose LLMs for their flexibility and abilities to categorize long-tail entities.

8 FUTURE WORK

In addition to the limitations and challenges described above, we also highlight other directions for future work.

1: More contextual features: AutoHistograms only categorizes based on entities, but there many features of interest are more subtle or contextual. For example, a sentence might be sexually explicit even if it does not contain any entities that are themselves explicit. Relatedly, other features such as tone or chattiness are higher-level than individual tokens. [31] have a method for clustering and annotating the text based on higher-level user-driven concepts; it would be interesting to find a way to automatically discover these concepts based on the dataset.

2: Different modalities: While AutoHistograms was built for text, the only text-specific aspect is entity extraction. For example, it could be extended to images by running an object detector over each image, and then clustering and labeling the detected image contents.

9 CONCLUSION

We present AutoHistograms, a tool that leverages LLMs and embeddings to create an interactive interface of automatically-generated distributions for data practitioners to analyze unstructured datasets. Through an observational study with 10 data practitioners, we validate that the tool can address targeted user needs such as summarizing datasets, identifying outliers and interesting slices of data, and testing hypotheses rapidly and interactively. Participants quickly identified the correct dataset topic, noticed a potentially-concerning asymmetrical data distribution, and found safety violations within the dataset. Finally, we summarize potential use cases and limitations of AutoHistograms described by these participants. Together, these findings suggest that advancements in LLMs can enable the development of sensemaking tools to better serve data practitioners.

ACKNOWLEDGMENTS

The authors wish to thank our colleagues at Google's People + AI Research Team for helpful feedback and discussions, especially Lucas Dixon, Andy Coenen, Martin Wattenberg and Fernanda Viégas.

REFERENCES

[1] [n. d.]. Data Quality for AI. https://www.ibm.com/products/dqaiapi
[2] [n. d.]. Introducing the Data Measurements Tool: an Interactive Tool for Looking at Datasets. https://huggingface.co/blog/data-measurements-tool
[3] [n. d.]. Know Your Data. https://knowyourdata.withgoogle.com/
[4] [n. d.]. Lilac: better data, better AI. https://lilacml.com/
[5] [n. d.]. Using machine learning to reduce toxicity online. https://perspectiveapi.com/
[6] Robert A. Amar, James R. Eagan, and John T. Stasko. 2005. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization (InfoVis)*. 111–117. https://doi.org/10.1109/INFVIS.2005.1532136
[7] Yannick Assogba, Adam Pearce, and Madison Elliott. 2023. Large Scale Qualitative Evaluation of Generative Image Model Outputs. *arXiv preprint arXiv:2301.04518* (2023). https://arxiv.org/abs/2301.04518
[8] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.
[9] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022.
[10] Richard Brath, Daniel Keim, Johannes Knittel, Shimei Pan, Pia Sommerauer, and Hendrik Strobelt. 2023. The Role of Interactive Visualization in Explaining (Large) NLP Models: from Data to Inference. *arXiv preprint arXiv:2301.04528* (2023). https://arxiv.org/abs/2301.04528
[11] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.
[12] Jaegul Choo, Changhyun Lee, Chandan K Reddy, and Haesun Park. 2013. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 1992–2001. https://doi.org/10.1109/TVCG.2013.212
[13] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling Language Modeling with Pathways. arXiv:2204.02311 [cs.CL]
[14] Mark Diaz, Sunipa Dev, Emily Reif, Emily Denton, and Vinodkumar Prabhakaran. 2023. Developing A Conceptual Framework for Analyzing People in Unstructured Data. In *Socially Responsible Language Modelling Research*. https://openreview.net/forum?id=QSPHfgw5fp
[15] Jesse Dodge, Ana Marasovic, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting Large Webtext Corpora: A Case Study on the Colossal Clean Crawled Corpus. In *Conference on Empirical Methods in Natural Language Processing*. https://api.semanticscholar.org/CorpusID:237568724
[16] Yanai Elazar, Akshita Bhagia, Ian Helgi Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Evan Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, Hannaneh Hajishirzi, Noah A. Smith, and Jesse Dodge. 2024. What's In My Big Data?. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=RvfPnOkPV4
[17] Christiane Fellbaum. 2010. WordNet. In *Theory and applications of ontology: computer applications*. Springer, 231–243.
[18] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for datasets. *Commun. ACM* 64, 12 (2021), 86–92. https://doi.org/10.1145/3458723
[19] Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2023. Repairing the Cracked Foundation: A Survey of Obstacles in Evaluation Practices for Generated Text. *J. Artif. Intell. Res.* 77 (2023), 103–166. https://doi.org/10.1613/JAIR.1.13715
[20] Philipp Grandeit, Carolyn Haberkern, Maximiliane Lang, Jens Albrecht, and Robert Lehmann. 2020. Using BERT for Qualitative Content Analysis in Psychosocial Online Counseling. In *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science*, David Bamman, Dirk Hovy, David Jurgens, Brendan O'Connor, and Svitlana Volkova (Eds.). Association for Computational Linguistics, Online, 11–23. https://doi.org/10.18653/v1/2020.nlpcss-1.2
[21] Maarten R. Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794* (2022).
[22] Timothy C. Guetterman and Michael D. Fetters. 2018. Two Methodological Approaches to the Integration of Mixed Methods and Case Study Designs: A Systematic Review. *American Behavioral Scientist* 62, 7 (2018), 900–918. https://doi.org/10.1177/0002764218772641 arXiv:https://doi.org/10.1177/0002764218772641
[23] Kostiantyn Kucher and Andreas Kerren. 2015. Text visualization techniques: Taxonomy, visual survey, and community insights. In *2015 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 117–121.
[24] Harsh Lara and Manoj Kumar Tiwari. 2022. Evaluation of Synthetic Datasets for Conversational Recommender Systems. *arXiv preprint arXiv:2212.08167* (2022). https://arxiv.org/abs/2212.08167
[25] Cassandra Overney. 2023. *SenseMate: An AI-Based Platform to Support Qualitative Coding*. Ph. D. Dissertation. Massachusetts Institute of Technology.
[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
[27] Mahima Pushkarna, Andrew Zaldivar, and Oddur Kjartansson. 2022. Data Cards: Purposeful and Transparent Dataset Documentation for Responsible AI. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT '22)*. ACM, 1776–1826. https://doi.org/10.1145/3531146.3533231
[28] Crystal Qian, Emily Reif, and Minsuk Kahng. 2024. Understanding the Dataset Practitioners Behind Large Language Model Development. *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems*.
[29] Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and Measuring the Geometry of BERT. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/159c1ffe5b61b41b3c4d8f4c2150f6c4-Paper.pdf
[30] Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M Aroyo. 2021. "Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. ACM, Article 39, 15 pages. https://doi.org/10.1145/3411764.3445518
[31] Zihan Wang, Jingbo Shang, and Ruiqi Zhong. 2023. Goal-Driven Explainable Clustering via Language Descriptions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. https://aclanthology.org/2023.emnlp-main.657.pdf
[32] He Zhao, Dinh Q. Phung, Viet Huynh, Yuan Jin, Lan Du, and Wray L. Buntine. 2021. Topic Modelling Meets Deep Neural Networks: A Survey. In *International Joint Conference on Artificial Intelligence*. https://www.ijcai.org/proceedings/2021/0638.pdf
[33] Lina Zhou, Shimei Pan, Jianwu Wang, and Athanasios V. Vasilakos. 2017. Machine learning on big data: Opportunities and challenges. *Neurocomputing* 237 (2017), 350–361. https://doi.org/10.1016/j.neucom.2017.01.026

A FEW-SHOT LLM PROMPTS

A.1 Labeling a set of entities

This is the prompt used to label a set of entities in Section 4.1. Note that `{entities}` is replaced with a comma-delimited list of entities to be labeled. The label is parsed by taking LLM's response up to the next new line character. Note that there is a "none" category for non-coheisive clusters.

```
Entities: rollouts, releases/rollouts, link-outs, rollout,
rollouts/releases, deliverables/dependencies
Label: release-related

Entities: unclear, 1265, good, expected, UpToDate, hot,
difficult, tomorrow, Russia
Label: none

Entities: Sleep, Making out, Shower, Morning, Funeral,
Driving, Eating
Label: activities

Entities: Man, Woman, Nonconforming
Label: genders

Entities: fabulous, outstanding, interesting, delicious,
beautiful, interesting, fascinating, awesome, wonderful
Label: positive adjectives

Entities: 1990s, 1970s, Early 2000s, 2000s, 1980s, 1920s,
1980, 1950s, Roaring Twenties
```

```
Label: decades

Entities: {entities}
Label:
```

A.2 Getting examplars of a category label

This is the prompt used to generate exemplars of a given label, used in Section 4.3. Note that `{label}` is replaced with the desired label. The resulting exemplars are parsed by taking LLM's response up to the next new line character, and splitting on commas.

```
Label: activities
Entities: Sleep, Making out, Shower, Morning, Funeral,
Driving, Eating

Label: decades
Entities: 1990s, 1970s, Early 2000s, 1980, 1950s, Roaring Twenties

Label: subjects
Entities: English, Post-modernism, Calculous, Robotics, Early
french literature

Label: genders
Entities: Man, Woman, Nonconforming

Label: {label}
Entities:
```

A.3 Evaluation word lists

These are the hardcoded word lists we used for the evaluation. These are word lists of specific features (religions, races, genders, professions, and toxic words) that are used frequently by dataset practitioners for dataset analysis. These are not dataset specific. While there are a few variations of these word lists, we use an aggregation of those in [16], [13], and [15]. We also include the toxic words from C4's blocklist [8]

```
hardcoded_wordlists = {
    'religions': ['atheism', 'buddhism', 'christianity',
    'hinduism', 'islam', 'judaism'],
   'race': ['asian', 'black', 'white', 'hispanic', 'indian'],
  'gender': ['she', 'her', 'hers', 'herself', 'he', 'him', 'his',
   'himself','they', 'them', 'their', 'theirs', 'theirself',
   'themself', 'themselves'],
   'professions': ['technician', 'accountant', 'supervisor',
   'engineer', 'worker', 'educator', 'clerk', 'counselor',
  'inspector', 'mechanic', 'manager', 'therapist', 'administrator',
   'salesperson', 'receptionist', 'librarian', 'advisor',
  'pharmacist', 'janitor', 'psychologist', 'physician', 'carpenter',
  'nurse', 'investigator', 'bartender', 'specialist', 'electrician',
   'officer', 'pathologist', 'teacher', 'lawyer', 'planner',
  'practitioner', 'plumber', 'instructor', 'surgeon', 'veterinarian',
  'paramedic', 'examiner', 'chemist', 'machinist', 'appraiser',
  'nutritionist', 'architect', 'hairdresser', 'baker', 'programmer',
  'paralegal', 'hygienist', 'scientist', 'dispatcher', 'cashier',
  'auditor', 'dietitian', 'painter', 'broker', 'chef', 'doctor',
   'firefighter', 'secretary'],
}
```

[8] https://github.com/allenai/allennlp/discussions/5056