Aunt Lily Can Say Her Visualizations: Directing Analysis, Design, and Storytelling in Natural Language (A Vision)

Zening Qu*
zqu@uw.edu
University of Washington

Fan Du
fdu@adobe.com
Adobe Research

Ryan A. Rossi
ryrossi@adobe.com
Adobe Research

Bill Howe
billhowe@uw.edu
University of Washington

ABSTRACT

We envision Lily, a flexible, lightweight, expressive, mixed-initiative authoring tool for anyone to direct data stories end-to-end — from insight-finding to storytelling — by saying words. We assume our vocabulary of visualizations is represented as an open repository of templates shared by authors. Our approach associates user queries with templates using learned embeddings, derives a set of variable bindings, and ranks the bindings by similarity with the user’s query. By supporting data binding and template customization via natural language, Lily inherits the expressiveness of the union of popular authoring tools like D3 and Data Illustrator, while relying more on the scale and quality of NLP models. We discuss how Lily can accelerate high- and low-level authoring workflows, facilitate coarse- and fine-grain customization, and accommodate different design styles and expression habits. We intend Lily to provide a natural means of data storytelling, for novices and experts alike.

Index Terms: Human-centered computing—Visualization—Visualization systems and tools

1 INTRODUCTION

“Could you help me learn Tableau?” requested Aunt Lily on a Sunday afternoon. She is a medication safety officer working with patient data. After attempting to create a dashboard similar to the ones left off by her former colleague (Fig. 1), and after much tutorial-cramming, she lamented, “The Tableau videos are helpful. However, it takes time and experience…”

Figure 1: An interactive dashboard like this requires not only design skills, but also technical know-how: how to build small multiples, how to add drop-down filters, how to show axis titles at the top of bar charts instead of at the bottom (the default formatting), how to publish dashboards as separate tabs to Tableau Server, and so on. Who knows all these execution steps by heart?

*This was Zening’s internship project at Adobe Research.

Menus and icons – our concrete, constrained, and oftentimes complicated way of communicating with our computers – are powerful but necessarily become more complex as tasks become more sophisticated. Into countless clicks and drags we translate our intents, no matter what they are. The more control we want, the more widgets we add. A plethora of tools surround visualization authors to help them analyze data and tell stories — data cleaning [17,18], analysis [42], insight finding [9], visualization design [22,34,36], interaction design [36], animation [43], and narration [35,41] — each offering its own expressiveness, constraints, and user interface. Much learning is needed.

Theoretically, Aunt Lily’s lack of design, analysis, and storytelling experience as well as technical know-hows could be compensated by natural language processing (NLP) and mixed-initiative recommendations. NLP could parse Aunt Lily’s intents, allowing her to direct where the project is going from where she is. And the unknown or unsaid design choices could be filled in or enumerated by machine recommendations. Compound NLP and recommendation in a mixed-initiative system, the author can gain more control by saying more, or give more control to the system by saying less.

Several systems have endeavored to use natural language for visualization authoring, either as the main channel of creation, or as a complementary to menus and icons [12,30,38,40,54]. However, the authors of these systems report it is not easy to create a good NLP-driven authoring experience. Why is the problem so hard?

NL-driven visualization authoring has to be both expressive and precise. It is hard to achieve both goals at the same time. Expressive means being able to render what the author has (vaguely envisioned) on her mind, including the macro and micro design choices, the styles and so on. Being precise means being able to respond to the subtle changes in the author’s expressions – parsing the author’s intents accurately. Inevitably, as expressiveness increases, the space of possible mappings from natural to some visual language grows exponentially, making it much harder for NLP to have an accurate understanding of the author’s intents.

Moreover, for a NL-driven authoring tool to be practically useful, the entire workflow (data analysis, visualization design, storytelling) should be readily accessible by natural language inputs. There should be no roadblocks. A single hurdle along the way can scare off masses of inexperienced potential authors. If Aunt Lily could create a visualization with a few words in a matter of seconds, but still needed to explore her data in Tableau or post-edit her data story in Adobe Illustrator, that would defeat the purpose of a fast and agile NL-driven authoring tool.

We need a better understanding of the natural language steering wheels along the authoring workflow. Therefore, we surveyed the literature and conducted interviews to crystallize the workflow — understand common steps taken by authors. Using this knowledge we designed Lily, a data analysis - storytelling workflow directed by natural language. We contribute:

1. A framework for characterizing and automatically recommending visualization templates, including their discoverability, customizability, constraints, and compatibility.
2. A design for Lily, an NL-driven authoring tool that accommodates the entire workflow from data wrangling to storytelling. By improving both expressiveness and accuracy along the entire authoring workflow, Lily brings us one step closer to the vision of custom visualizations created for and by the masses.

2 Design Goals

We designed Lily to achieve the FLEX goals:

FLEXIBLE Recognize natural language instructions of various types, succinct or elaborate, at macro and micro levels, expressed with different word choices and styles, and tolerate errors and ambiguities.

LIGHTWEIGHT Provide a simple user interface that does not demand much learning about the interface itself. Avoid overdemanding the author with the system input, or overwhelming them with the output.

END-TO-END Support the entire data-to-story authoring workflow.

EXPRESSIVE Support diverse visualization and data story designs. Allow authors to have control over the design details.

3 User Experience (UX) Design

We show an end-to-end workflow in which Aunt Lily overviewed data, customized design details, searched for an insight, and commanded a story to be told (Fig. 2a-d). These macro and micro instructions all go through a single NLI (Fig. 3).

It had been a year since COVID-19 broke out in the US. Aunt Lily wanted to pull the latest data, reflect, and write a letter to strengthen her colleagues, who had been fighting unprecedentedly hard and long. She selected a dataset from CDC.

At first, she didn’t know what to say. The mixed-initiative agent suggested an ice-breaking query: “What's in my data?” She clicked on it and saw a variety of data overviews (Fig. 2a), each covering a distinct subset of data attributes, exposed as underlined links in the automatically generated chart titles. If clicked, an attribute will become a new query.

But Aunt Lily didn’t want to fixate on any specific attributes yet. Instead, she loosely expressed her interests as “cases in more map designs.” Now the only attributes visualized are those matching “cases,” plus the auxiliary geographical fields required by “maps.” Meanwhile, more diverse map designs are shown: besides the choropleth map, there is also a hexbin map, a bubble map, and a spike map. The design names show up as tags beside the chart titles. They can also form future queries.

The colors were also diversified: cumulative confirmed cases in sequential shades of blue, cumulative confirmed cases in diverging shades of pink and teal, and daily new confirmed cases in orange. Such diversification follows Qu and Hullman’s consistency principles [32], which says different data should wear different colors. It also reveals many color scheme options to the author. For instance, Aunt Lily picked “pink and blue” (Fig. 2b).

The spike map made Aunt Lily quite concerned about New York.
She revised her query to "cases most affected state". Because the query conveyed an intent to find extremes, the result visualization contained highlights – the mark representing the highest value was emphasized whereas all others were dimmed. Because the query said 'state', all charts were aggregated at the state level. The word 'most' in the query was retold as rankings and highest in the chart titles. The synonyms provide a flexible way to explore data by a statistical insight.

In the timelines, Aunt Lily saw that California is surpassing New York in recent death counts. She held the microphone and said: "Highlight California and New York and write 'It's painful to see California surging after New York. My colleagues, let’s stay alert.'" And the system updated the highlights and annotated all charts (Fig. 2c).

Then Aunt Lily said: "Write a letter using the line chart, and some other charts we've seen." Now the annotated timelines became a part of a scrolly, which was automatically pre-populated by the year round radial, the hexbin map, and the spike map (serving as a background image for the title). The year round radial was in the letter because Aunt Lily had clicked its storytelling icon during "What's in my data?" The hexbin map and the spike map were added by the system because Aunt Lily had said "use ... some other charts" but did not specify which ones (Fig. 2d). Aunt Lily went on to edit the titles, the captions, and the body text of her letter. She could add, remove, replace, and reorder the visualizations in her letter by saying queries like "replace the hexbin map with the one that had blue bubbles" – if she could only recall certain chart names, data attributes, or colors. Alternatively, she could collect story pieces from her query history, and cut and move pieces in her letter. Finally, Aunt Lily signed her letter and sent it.

4 Envisioning the System

L.U.Y consists of a user interface, a natural language processing module, a recommender, a renderer, and a collaborative platform that serves repositories of data and insights, and templates and components (Fig. 4). A template is a data story or visualization with customizable data and design components (e.g., rebind data attributes, hide axes, replace a drop down menu with a slider). Any data story or visualization that can be represented as a string (e.g., HTML and D3 [3], Jupyter Notebook and Altair [46], R Markdown and ggplot2 [49]) may become a template in L.U.Y. When all components are specified, a template becomes a renderable specification. Each specification is associated with a string signature, which can be any appropriate linearized representation. Each signature is encoded using a deep language model (e.g., GPT-3 [4], CPM-2 [55], BERT [10], ELMo [31]), possibly fine-tuned on available examples (NL2VIS specifications, revisions, constraints, and comments, scrapped or synthesized [24]), building on recent approaches [5].

Query To begin, the user speaks or types a natural language query. The query may refer to the resources on the collaborative platform: data tables, attributes, values, insights, and transformations; design templates, components, and customizations. The same language model encodes both queries and specification signatures, making the distance between them measurable.

Recommend The recommender searches and ranks the top k specifications that best match a query given its context (Fig. 4). First, it filters templates with the data constraints (e.g., "cases") and design constraints (e.g., "red map") in the query. For each template, the recommender enumerates all possible customized specifications based on the query. We bind attributes to template variables according to typical visualization principles using automated constraint satisfaction tools [29]. Each attribute is encoded, and the distance from the encoded attribute and the encoded query is used as a soft constraint to guide the solver [29]. The result is all possible variable configurations that satisfy the template and preferring attribute similarity.

4.1 System Discussion

We intend our design to enable study of the limits of NLP methods as well as the capabilities. To that end, we emphasize breadth in our requirements.
Queries and Intent  

In all NLP Vis systems, the goal is to infer an author’s intent (what they are trying to achieve) given their query (the actual sequence of words they typed or spoke) [45].

Rather than try to derive a vocabulary of intents from a small set of actions supported by a system, we treat every stemmed n-gram as a potential intent, and every possible instantiation of a template as a potential recommendation. While we avoid needing to design a universe of discourse ourselves, we are now more dependent on the amount and quality of the data. However, Lily can use pre-trained models at first while data is collected, and then apply joint learning techniques [14, 53] to learn embeddings from associated pairs of templates and queries, validated by author selections.

To develop the initial vocabulary, we anticipate scraping visualization textbooks, papers, tutorials, and forums for the template construction.

To enable more use of their data, and template contributors who want to popularize their novel chart invention, distinct users: data analysts (our primary focus), data contributors, and analysts and contributors themselves, we are more dependent on the amount and quality of the data. However, Lily can use pre-trained models at first while data is collected, and then apply joint learning techniques [14, 53] to learn embeddings from associated pairs of templates and queries, validated by author selections.

Template Registration  

We anticipate a variety of potential utterances, all of which can potentially be interpreted in our general approach of matching utterances to templates.

- a (partial) specification ("write a letter using...”),
- a direct revision ("change colors to pink and blue")
- a constraint (e.g., "axis titles on top")
- a comment (e.g., "it’s too colorful")
- a concept (e.g., "overview", "surprise me!", "correlation", "normal distribution")
- a reference (e.g., "cases", "the one that had blue bubbles") or a question (e.g., "what’s in my data?")

Improving Data Quality with Computed Labels  

To improve the matching of variables to data columns, we can compute insights [20] on all available data. Data facts as implemented in Cui et al. can also be wrapped as functions [7].

To improve the matching of templates to queries, we can apply weak supervision by writing multiple simple yet unreliable data labels [33]. For example, templates from a particular website with a name that includes `scatter` can be tagged as a scatter plot. In this way, data quality can be improved incrementally and continuously by adding new labels.

Analysts and Contributors  

We intend Lily for three communities of users: data analysts (our primary focus), data contributors who want to enable more use of their data, and template contributors who want to popularize their novel chart invention, distinct styles, animated transitions, or new interaction techniques. We intend to support analysts of any skill level, looking for opportunities to automate decisions that complicate typical tools.

Template Registration  

When a new template is contributed to the platform, we ask the contributor four questions:

1. Discoverability: Please describe this template in your own words. What data story does it tell? This will make it easier for other users to find your template.
2. Constraints: What kind of insights, data distributions, cardinalities, themes etc. is this template best for? Are there any occasions or situations where using this template would be inappropriate?
3. Customizability: What parts of the template can be changed by the user (e.g., variables, constants, APIs)?
4. Compatibility: Can this template be repeated or linked or combined with other templates or interactive widgets?

5 Related Work  

Manual Visualization Authoring Tools  

Most of the current business intelligence [23, 28, 42] and web analytics [1, 13] offerings focus on a manual workflow for data exploration and visualization authoring, where analysts need to select which variables to explore, decide what kind of visualization charts to use, inspect if useful insights exist, and repeat. These tools are powerful yet too sophisticated for non-exerts who have limited data science knowledge or graphic design skills.

Automated Analytics Tools  

More recently, automated data analytics tools have gained great traction [21]. Data science models have been proposed to automatically select interesting data variables [8, 9, 47, 50, 51] or generate appropriate visual representations [15, 25, 29]. Among commercial products, Tableau offers a "Show Me" panel, which automatically recommends chart types based on users’ variable selections [26]. These automated analytics tools can provide helpful guidance, but they lack an input channel for users to communicate their intents so as to get desired results quickly.

Search Analytics Tools  

During the last decade, data analytical workflows driven by natural language queries have emerged [52, 54]. Adobe Analytics pioneered a technology prototype called DataTone that allows people to retrieve statistical data facts from a dataset in natural language [2, 12]. Tableau and ThoughtSpot also released similar “Search Analytics” products [27, 38, 44]. While these tools provide an innovative natural language interface for users to specify their analytical intents, they often require the users to be familiar with the data and specify concrete data queries, such as "Show me a bar chart of sum of order total value by geo cities". However, when authoring a data story, non-expert users often only have vague ideas or questions, such as "How is the revenue?". It is difficult for them to map their ideas to the concrete queries. To address this challenge, our system uses intent recognition techniques to infer users’ analytical interests from ambiguous or partial queries and uses recommendation models to generate complete analytics specifications.

6 Two Case Studies  

We show how our proposed system design works in two extreme cases: using Lily as a “one-button machine” that generates data stories with minimal human intervention, and using Lily for elaborate, precise control. The point is, Lily works in both scenarios, which shows it flexibility. And being able to recreate the expert dashboard in Fig. 1 further demonstrates Lily’s expressiveness.

Case 1: One-Button Machine  

Uncle Nelson is a busy user. He has the same data as Aunt Lily, but has no time for exploration, customization, or composition. He wants a data story and he wants it now! He says: “data story”. And Lily returns a poster, a story-line, a data video, and so on. The poster compares COVID-19 case counts in various cities in a small multiples design. Data attributes were binded and data insights were annotated by Lily. The original poster template had won an Information is Beautiful award [19]. The storyline used a Calliope [39] template, showing insights captured by DataShot [48]. The data video template was contributed by editors at Johns Hopkins University. Uncle Nelson prints out the poster and heads to his meeting.

In this case, Lily detects the intents “data”, “story”, and “data story”, which is more specific and likely to be accurate than “data” and “story” alone. Since the most likely intent is at “data story” level, Lily’s recommender decided to diversify at the next-level specificity – the genre of the story. Awards and the popularity affected the ranking of the binded and customized templates. Data attributes, being top-level and unspecified in the query, is also diversified. If Uncle Nelson were to implement these data stories on the
his own, he would need to start with researching data story templates, and handle the data binding and customization on his own, which he would not have time for.

Case 2: Precise Control Uncle Rodney is an expert user. He and Aunt Lily sat down together to replicate the dashboard by their ex-colleague (Fig. 1). To achieve this goal, they need precise control over the visual encodings, the formatting of axes and fonts, the layout, the colors of the alternating rows, and so on. Our supplementary material records 19 queries they said in order to replicate the dashboard. When we compared their queries with the equivalent 19 queries they said in order to replicate the dashboard, we became more hopeful – systems like Lily, if implemented well, could alleviate the pain of technical know-hows for many people.

Acknowledgments
We thank Aunt Lily, Uncle Rodney, and Uncle Nelson for letting us using their names in our paper. We thank FlatIcon.com for the icons 🎨🌟🎨, and OpenMoji.org for 🌍.

References


Supplementary Material
Aunt Lily Can Say Her Visualizations

A Expert Dashboard in Lily and Tableau

Tableau: We re-creating the expert dashboard in Fig. 1 in Tableau. We take the easiest route we know of. We omit the googling about how to do certain things in the UI in these notes.

Novice: How a novice author such as Aunt Lily could achieve the same result using Lily.
Expert: How an expert author could do the same.

A.1 Specify X Y Encoding

Tableau: Find each of the eight measures – # Enc, # Orders, Administrations, Admin Dose, Medication Scan Required..., Med Scanned At Admin, Med Scan at Admin % ..., and Age At Encounter – in the ‘Data’ panel by eyeballing or searching. Drag each measure to the end of the Columns shelf. Find ‘Day of the Week’ and drag it to the ‘Rows’ shelf.

Novice: “Create a bar table that look like those in Tableau, using Day of the Week and # Enc, # Orders, Administrations, Avg. Admin Dose, Medication Scan Required..., Med Scanned At Admin, Med Scan at Admin % ..., and Avg. Age At Encounter.”
Expert: “Tableau-style concatenated bar charts. Day of the Week by # Enc, # Orders, Administrations, Avg. Admin Dose, Medication Scan Required..., Med Scanned At Admin, Med Scan at Admin % ..., and Avg. Age At Encounter.”

A.2 Add Colors

Tableau: Make sure ‘Marks’ is set to ‘All’. Drag ‘Measure Names’ to ‘Color’.

Novice: “Add colors." "Color each bar chart differently." "Color the distinct fields."”
Expert: “Color by measure names." "Color the quantitative fields."”

A.3 Take Averages

Tableau: For two measures (‘Admin Dose’, ‘Age At Encounter’), click on their drop-down arrow, ‘Measure (Sum)’, ‘Average’.

A.4 Add Labels


Novice: “Show the numbers in bold. Don't show the .00s.”
Expert: “Label the values in bold. Don’t show decimals.”

A.5 Hide Axis Numbers and Ticks

Tableau: For each of the eight measures, right click on the axis title, select ‘Edit Axis...’. In the popup window, set ‘Sort By’ to ‘Manual’, and sort the days from Sunday to Saturday.

Novice: “Sort the week on Sunday.”
Expert: “Sort Sunday to Saturday.”

A.6 Sort from Sunday to Saturday

Tableau: Right-click on the white space in the worksheet, select ‘Format...’. Click on the ‘Sort Order’ icon. Under ‘Default’, click on the ‘Header’ dropdown menu and toggle the ‘B’ icon to bold the header text.

Novice: “Make # Enc, # Orders, Administrations, Avg. Admin Dose, Medication Scan Required..., Med Scanned At Admin, Med Scan at Admin % ..., and Avg. Age At Encounter.”
Expert: “Bold all axis titles.”

A.7 Bold All Axis Titles

Tableau: Right-click on the white space in the worksheet, select ‘Format...’. Click on the ‘Format Font’ icon. Under ‘Default’, click on the ‘Header’ dropdown menu and toggle the ‘B’ icon to bold the header text.

Novice: “Make # Enc, # Orders, Administrations, Avg. Admin Dose, Medication Scan Required..., Med Scanned At Admin, Med Scan at Admin % ..., and Avg. Age At Encounter.”
Expert: “Bold all axis titles.”

A.8 Alternate Row Background Colors


Novice: “Blue and white rows.”
Expert: “Alternate blue and white bands between rows.”

A.9 Reduce Lines


Novice: “Keep grid lines etc. to the minimal.” “Hide all non-essential/unnecessary/auxiliary light gray lines.”
Expert: “Reduce grid lines.” “Minimalistic flat design.” “Hide the row and column dividers and the grid lines.”

A.10 Move Axis Titles to the Top


Novice: “Move # Enc, # Orders, Administrations, etc. to the top.”
Expert: “Show axis titles on top.” “Top axis titles.”

A.11 Add Dropdown for Time

Tableau: Drag ‘Month, Year of Med Order Instant’ to ‘Filters’ card. In the popup window, select ‘Month/Year’. Click ‘Next >’. Select multiple months by checking the boxes. Click ‘OK’. In the ‘Filters’ card, click on the dropdown arrow of ‘MY(Month, Year of Med Order Instant)’ and select ‘Show Filter’. In the filter card, click on the dropdown arrow and select ‘Multiple Values (dropdown)’. In the popup window, set the title as ‘Month, Year of Med Order Instant’. In the filter card, click on the dropdown arrow and select ‘Multiple Values (dropdown)’. Now the filter card appears as a dropdown menu.

Novice: “Allow people to select multiple Month, Year of Med Order Instant.”
Expert: “Add a multi-value dropdown menu for Month, Year of Med Order Instant.”
A.12 Add Dropdown for Antibiotics

Tableau: Drag ‘Ordered Med Name’ to ‘Filters’ card. In the popup window, click the ‘Use all’ radio button. Click ‘OK’. In the ‘Filters’ card, click on the dropdown arrow of ‘Ordered Med Name’ and select ‘Show Filter’. In the filter card, click on the dropdown arrow and select ‘Single Value (dropdown)’.

Novice: "Let people change Antibiotics.”

A.13 Hide Color Legend

Tableau: Delete the color legend for ‘Measure Names’ by clicking the ‘X’ icon on the upper-right.

Novice: “Don't show which color is which field.”
Expert: “Hide color legend.”

A.14 Edit Dashboard Title


Novice: "Dashboard." Then directly edit the dashboard title to ‘Medication Administration Details - Weekday/Day of Month by Med’. Select 18 pt bold font, blue color, align center.
Expert: "Make it a dashboard titled ‘Medication Administration Details - Weekday/Day of Month by Med’, 45 pt bold, blue color, align center.”

A.15 Edit Visualization Title

Tableau: Double click on the default worksheet title ‘Sheet 1’ and name the sheet ‘Breakdown by Day of the Week’. Select Tableau Bold 36 pt font, black color, align left. Right click on the sheet title, select ‘Format Title’, under ‘Title Shading’ select a light gray color.

Novice: Directly edit the visualization title ‘Breakdown by Day of the Week’. Select 36 pt bold font, black color, lightgray background, align left.
Expert: "Title the bar charts ‘Breakdown by Day of the Week’, 36 pt bold, lightgray background, align left.”

A.16 Hide the Y Axis Title

Tableau: Right click on the rows axis title ‘Day of the...’, select ‘Hide Field Labels for Rows’.

Novice: "Erase the text Day of the...", select ‘Hide the axis title Day of the Week’.
Expert: "Hide the axis title Day of the Week.”

A.17 Edit Dashboard Layout


Novice: Drag and drop the elements in the dashboard and the X Y W H layout boxes should appear like in Adobe Photoshop.
Expert: "Bring out the layout panel. or "Edit dashboard layout. or simply "Layout." Or do this in queries "Make the dashboard 1600px wide and 500px tall." “The dashboard title should be 1600px wide and 50px tall, position it at 0,0.” “Position the two dropdown menues under the dashboard title. Each should be 300px wide and 50px tall, align left.”
"Put the bar chart at X=10, Y=120. It’s size should be 1580x350." or "Put the bar chart underneath the dropdown menues. It should take up the entire width of the dashboard“.

A.18 Add Timestamp

Tableau: Make sure the dashboard is selected. In the ‘Objects’ panel, click the ‘Floating’ toggle, and drag ‘Text’ to the dashboard. In the popup window, write ‘Updated 6/7/20’ and bold ‘6/7/20’. Select the text and click on the ‘Layout’ tab. Set its ‘Position’ to ‘x = 1490’, ‘y = 10’, and set its size to ‘w = 100’, ‘h = 50’.

Novice: Click the textbox icon and add the timestamp via direct manipulation.
Expert: "Write ‘Updated 6/7/20’ and bold 6/7/20, 36 pt, right aligned, and center aligned to the dashboard title.”

A.19 Publish as Tabs

Tableau: Double-click ‘Dashboard 1’ at the bottom of the window and rename the dashboard to ‘Med Admin Details - Weekday/Day of Month by Med’.

Click on the ‘Server’ menu and select ‘Publish Workbook...’. In the popup window, select this dashboard along with three others and check the ‘Show sheets as tabs’ checkbox. Click ‘Publish’.

(Recreated with Tableau Desktop Professional Edition, 2020.3.10)

Novice: “This is a tab. The other tabs should be Med Admin Details - by Med Name and Med Admin Details - by Order. Publish these three tabs together.”
Expert: "Add in Med Admin Details - by Med Name and Med Admin Details - by Order as separate tabs in this dashboard. Publish.“