Answering Questions about Charts and Generating Visual Explanations: Supplemental Materials

Dae Hyun Kim Stanford University dhkim16@cs.stanford.edu Enamul Hoque York University enamulh@yorku.ca

Maneesh Agrawala Stanford University maneesh@cs.stanford.edu

FORMATIVE STUDY

Words for Referring to Visual Features

From the 277 visual questions we collected prior to the formative study, we identified the words that people use to refer to the visual features of the charts. We complied these into word lists and use them to detect mark words, visual attribute words and visual operation words in Stage 2 of our pipeline. Here, we include the word lists that we compiled (Figure 1).

Additional Analysis

In addition to the analysis of how often people ask visual/nonvisual questions, ask lookup/compositional questions, and provide visual/non-visual explanations, we further analyzed the questions from the formative study to determine which visual elements of the charts people referred to when asking questions or explaining their answers visually. Furthermore, we analyzed if people provide visual explanations when answering visual questions.

Visual Questions 43% of the visual questions included mark words (e.g. 'bar', 'line'). More visual questions referred to the color attributes of the marks (54%) than the length attributes of the marks (22%). 22% of the questions referred to the elements on the axes (e.g. the axis itself, label, ticks).

Visual Explanations 87% of the visual explanations included mark words. Unlike for visual questions, more visual explanations referred to the dimension of the marks (42%) than to the color attributes of the marks (30%). 13% of the questions referred to the elements on the axes.

Explanations to Visual/Non-Visual Questions 60% of the explanations to the visual questions were visual, whereas 50% of the explanations to the non-visual questions were non-visual. Visual explanations were slightly more common when the questions were visual, and for both visual and non-visual questions, people provided visual explanations at least half of the time.

ADDITIONAL DETAILS FOR EXPLANATION GENERATION

We generate the visual explanations from the lambda expressions via using a series of regex rules in Stage 3. Here, we provide more information about the rules used for explanation generation. For specific implementation details, please refer to the released code.

Natural Language Conversion Rules

In Stage 3 step 1, our pipeline converts lambda expressions to natural language using a small set of rules. Figure 2 shows a set of rules for this process. Please refer to the released code for specifics and precedence.

Redundancy Cleanup Rules

We remove two types of redundancies during redundancy cleanup (Stage 3 step 3): (1) repeated mentions of field names or values (e.g. "*age' of the greatest 'age*"') or (2) unnecessary mention of field or the word '*data'* (e.g. "*Country*' '*China*"'). Figure 3 lists some of the regex used in this process. For the specific regex we use, please refer to the released code.

Encoding Application

In Stage 3 step 5, our pipeline applies the encodings to convert references to field names and field values in explanations into the visual attributes of the marks to generate visual explanations.

Choosing Color Words Whereas people may use various color names to describe a color they see, explanations need to be clear and a small set of common color names is all that is needed to distinguish the marks. Because the common color names are better spread out throughout the hue-space than in the RGB space, we use the HSL color space to assign names to colors. In comparison, we use RGB color space for matching color words to colors in the chart because RGB color space has a naturally defined metric that allows distance comparisons to different colors used in the chart, whereas the HSL color space does not. We split the hue space into smaller slices according to the color names given by WorkWithColor.com [1]. We split the color ranges of the half-colors (e.g. red-orange, yellowgreen) into two halves and merged them with the closest hue range, resulting in a total of eight colors (Figure 4a). For lightness, we named colors with lightness greater than 87.5% as 'white' and colors with lightness less than 12.5% as 'black'. We further split the lightness space and add the adjective 'light' when the lightness is between 75% and 87.5%, and 'dark' if it is between 12.5% and 25%. For saturation, we name colors 'gray' if it has saturation less than 12.5%. If the light or dark

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). *CHI '20, April 25–30, 2020, Honolulu, HI, USA.* Copyright is held by the author/owner(s). ACM ISBN 978-1-4503-6708-0/20/04. http://dx.doi.org/10.1145/3313831.3376467



Figure 1. Word lists used in Stage 2 of our pipeline for marks of type 'bar' and 'line' (Extension of Figure 6 in the main paper). The list of words referring to these marks (cyan), the list of words for referring to the visual attributes of the marks (orange) and the list of words for representing operations on the marks (green).

[Lookup]	R [property1].arg2	propertyl of arg2
[Type]	R [type1].arg2	arg2 (no-op)
[Row]	Row	data
[Argmax]	$\arg\max(\arg1, \mathbf{R}[\lambda x(\arg2.x)])$	arg1 with the greatest arg2
[Argmin]	$\arg\min(\arg1, \mathbf{R}[\lambda x(\arg2.x)])$	arg1 with the smallest arg2
[Max]	max(arg1)	maximum arg1
[Min]	min(arg1)	minimum arg1
[Greater/Equal]	>=(arg1)	greater than or equal to arg1
[Greater]	>(arg1)	greater than arg1
[Less/Equal]	<=(arg1)	less than or equal to arg1
[Less]	<(arg1)	less than arg1
[Difference]	-(arg1, arg2)	difference between ${\tt arg1}$ and ${\tt arg2}$
[Sum]	sum(arg1)	sum of arg1
[Count]	count(arg1)	number of arg1
[Average]	avg(arg1)	average of arg1
[And]	and(arg1, arg2)	argland arg2
[Or]	or(arg1, arg2)	argl or arg2
[Field Value]	field1.value2	field1 value2
[Lambda]	$\lambda x(arg1.x)$	argl(no-op)
[Reverse]	R [arg1]	argl(no-op)

Figure 2. Conversion rules from lambda expressions to natural language (Extended version of Figure 9 in the main paper). The first column shows the name of the rule, the second column shows the lambda expression and the third column shows the corresponding natural language expression.

Redundant Expression	Cleaned Expression
[arg1](of)[arg1]	[arg1]
[arg1] with the greatest [arg1]	the greatest [arg1]
[arg1] with the smallest [arg1]	the smallest [arg1]
[arg1] of data	[arg1]
[field1][value-of-field1]	[value-of-field1]

Figure 3. Redundancy cleanup rules for explanations. The first column shows the original redundant expression, and the second column shows the cleaned-up result.

shade of the color is often interpreted as a different color, we specially defined the color name (e.g. *'brown'* for *'dark orange'*). Figure 4b shows the split for orange.



Figure 4. How the colors are named in the HSL space. (a) We split the hue space into eight color ranges. The example colors represent colors in the center of the range with 100% saturation and 50% lightness. (b) For each hue value, we split the saturation and lightness space into black, white, light and dark versions of the color, and the color itself. Here, we exemplify this with the color orange (with hue value 30° . We use 'brown' instead of 'dark orange' because it is a more commonly used color name.

Word Rearrangement Simply applying the encodings may result in natural language expressions that could be made smoother by rearranging the words. For example, "*length of 'China''* can be smoothened by adding the mark word into "*length of the bar for 'China''*. In order to do so, we apply a series of regex rules to the resulting explanations (Figure 5). Please refer to our code for the implementation details as well as the exact precedence.

ADDITIONAL RESULTS: EXPLANATIONS

Because the explanations generated by our system in stage 3 are templated conversions of Sempre [4, 3]'s lambda expressions [2], the generated explanations are reasonable as long as the lambda expression output corresponds to meaningful operations. Here, we share some less meaningful explanations generated because the original lambda expression did not represent meaningful operations on charts (Figure 6).

Original Expression	Reworded Expression
length of [value1]	length of the bar for [value1]
height of [value1]	height of the [bar/line] for [value1]
[arg1] of the length	[arg1] of the bar with length
[arg1] of the height	[arg1] of the [bar/line] with height
greatest the length of the	longest
smallest the length of the	shortest
greatest the height of the	[tallest/highest]
smallest the height of the	[shortest/lowest]
length with	length of the bar with
height with	height of the bar with
greatest the length	longest length
smallest the length	shortest length
greatest the height	[tallest height/greatest height]
smallest the height	[shortest height/smallest height]
number of the length	number of bars with length
number of the height	number of [bars/points] with height
[bar/line] (of) [value1]	[bar/line] for [value1]

Figure 5. Word rearrangement rules for visual explanations. First column shows the original expression that can appear in the visual explanations and the second column shows the reworded result. The bracketed expressions with two options indicate word choice when the chart is a bar chart (left) and when the chart is a line graph (right).



Figure 6. Examples of less meaningful explanations generated by our pipeline. The first row shows the question, and the second row shows the answers generated by our system (red indicates incorrect) and the correct answer. The third row shows the lambda expression generated by Sempre and the last row shows the explanation generated by our system.

For Q1, our pipeline generates an explanation that simply states the answer 'Glabron' without any operations on it. Observing the labmda expression, our system finds the row of the underlying data table with the 'variety' value equal to 'Glabron', and obtains the 'variety' value of that row, which is equivalent to just reporting 'Glabron'. Because the operations in the lambda expression are very redundant, our system results in removing all the redundant operations and ends up giving the meaningless explanation.

For Q2, our pipeline generates an explanation with the word 'index', which is not defined with respect to the chart. This is because the lambda expression operates on the underlying data table and not the chart itself. The lambda expression indicates that it read the variety value of the last row of the table, which has no correspondence in terms of the chart because the ordering of rows in the table does not necessarily match that of the ordering of the 'varieties' We leave better incorporation of such table-specific operations as future work.

ADDITIONAL RESULTS FOR USER STUDY

In addition to the Likert scale measurements of transparency, trust and usefulness, we also measured how accurately participants determined the correctness of the provided answers, and how quickly the participants so (Table 2 in the main paper).

Accuracy

We did see higher accuracies when we provided answers and explanations generated by our system (98.8% with visual explanations and 95.0% with non-visual explanations) than when we provided answers generated by humans (91.3% with explanations and 87.5% without explanations). While this could be due to our explanations, this could also be due to the wrong answers by our system being more conspicuous than wrong answers generated by humans. Further study is required to determine the contributions of these factors.

Time Measurements

Although we measured time taken to determine the correctness of the provided answers, we did not see a significant improvement in completion times when we presented our visual explanations ($\mu = 26.7$ s, $\sigma = 30.5$ s) compared to when we presented no explanation ($\mu = 26.2$ s, $\sigma = 28.3$ s, t(157) =0.11, p = 0.46), human-generated explanations ($\mu = 26.0$ s, σ = 27.5s, t(157) = 0.16, p = 0.44), or our non-visual explanations ($\mu = 23.7$ s, $\sigma = 21.2$ s, t(157) = -0.72, p = 0.76). Instead, we saw large variations in completion times in all conditions. This is probably because we did not instruct the participants to optimize for time. Other factors could be because the time required to perform the operations to confirm the answers was much greater than the time required to parse the provided answers and explanations with respect to the provided charts. Additional studies could help understand how explanations affect the speed at which people parse information.

REFERENCES

- [1] 2019. WorkWithColor.com. "Color Names". (2019). Retrieved December 30, 2019 from http://www.workwithcolor.com/color-names-01.htm.
- [2] P. Liang. 2013. Lambda Dependency-Based Compositional Semantics. arXiv preprint arXiv:1309.4408 (2013).
- [3] Panupong Pasupat and Percy Liang. 2015. Compositional Semantic Parsing on Semi-Structured Tables. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Vol. 1. 1470–1480.
- [4] Yuchen Zhang, Panupong Pasupat, and Percy Liang. 2017. Macro Grammars and Holistic Triggering for Efficient Semantic Parsing. In Proceedings of the 2017 *Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1214-1223.