Visualizing World Color Survey Dataset

Yi (Joshua) Ren

Abstract—To be done...

Index Terms—WCS dataset, visualization, distribution, quantitive metrics

1 Introduction

To be done...

The general goal of WCS project is to provide insights of language evolutionary process, but it is hard for the researchers to draw any conclusions directly from the raw data, which is stored in the text form. Hence an appropriate VAD toolkit would be helpful for the researchers who are interested in WCS dataset, which is the goal of our paper.

2 Data, Task and Visualization

In this section, we first provide a general overview of the WCS project: what is the goal of the project, who are the participants of it, what task are they asked to do and what kind of data do they generate. Then we will introduce the VAD principles and software tools we used in this paper. As the raw WCS dataset is stored in the form of text, which is hard for researchers to get intuitions, the data visualizations we proposed can scaffold their future research.

2.1 WCS Project and the Dataset

To study the language evolutionary process, many researchers narrow down their focus from the entire language (which is used to explain all possible concepts in the world) to some simple but representative tasks. One of the well known task is the world color survey (WCS) project [7], in which volunteers from different regions are asked to describe different colors using their first languages.

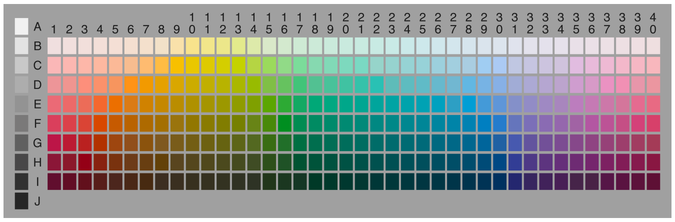


Fig. 1. Standard Munsell color card.

Specifically, they use a standard Munsell color card in this task, as illustrated in Figure 1. In this color card, 330 different colors are arranged in two groups: the left column contains blocks with 10 different gray scales (brightness, denoted by $\mathbb{B} = \{A,...,J\}$) and the right region contains 320 different colors. In the right region, there are 40 columns labeled as $\mathbb{H} = \{1,...,40\}$ representing different hues, and 8 rows with $\mathbb{B}_{color} = \{B,...,I\}$ representing different brightness. Note that in the right region, the brightness only ranges from B to I[1]. All of these colors are **randomly** indexed with a chip number ranges from 1 to 330. Hence we can refer to a specific color whether using its index (denoted by $idx \in \{1,330\}$) or by a tuple $(h,b) \in \mathbb{H} \times \mathbb{B}$, e.g., $(A,01)$[2].

The volunteers are selected from 110 different regions. In other words, the dataset has 110 different language types. There are on average 24 native speakers for each language. The gender and the age of all the participants are stored in *spkr.txt*. To get knowledge from the ancient languages, some of the participants are carefully selected from those preindustrialized cultures that had limited contact with modern, industrialized society.

Each participant will first read the instructions, and then name different colors on the Munsell using their mother language. As these languages are using different alphabets, the researchers use abbreviations (one or two capital letters) for each phrase[3].

2.2 Data Visualization Toolkit for WCS dataset

Data visualization usually acts as a scaffold for the researchers who want to find some trends from the raw data [5]. It is also important when we want to introduce our work to other researchers. As all of the important information of the WCS project is stored in text form, an appropriate VAD toolkit is necessary. Combining the research requirements in evolutionary linguistics and VAD principles, we propose a series of figure designs (so as a software toolkit) to assist a broad range of downstream tasks.

Specifically, we focus on the following three aspects of this dataset:

- Data visualization for the distribution of the participants;
- Data visualization for the naming of SINGLE language;
- Data visualization for the calculated quantitive metrics.

The motivations, detailed designs and some application examples of them are expatiated in the rest of the paper. Generally speaking, our paper aims to provide researchers a toolkit to help them generate appropriate visualizations during their whole project: from the brain storming phase to writing report. The toolkit is written in Python, using two common packages, i.e., matplotlib[4] and seaborn[5]. The readers can download the source code and modify any settings as they wish: they can use it as a starting point of an interactive design, or directly draw their own figures and put them on their report.

3 Visualization Design for Participants' Distributions

3.1 Overview of the task

When analyzing the WCS dataset, the first thing to consider is whether the selection of participants is unbiased. Usually, we wish the participants follow an uniform distribution on both gender and age, both inside each language and across all possible languages. If there are

- *renyi.joshua@gmail.com*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

[1] That is because A is pure white and J is pure black.

[2] The indexing of the color card can be found in the *chip.txt* file in WCS dataset.

[3] For example, in language 1, *gbanagbana* is mapped to *GB*. The mappings are stored in *dict.txt*.

[4] The website of this package is: `https://matplotlib.org`

[5] The website of this package is `https://seaborn.pydata.org`

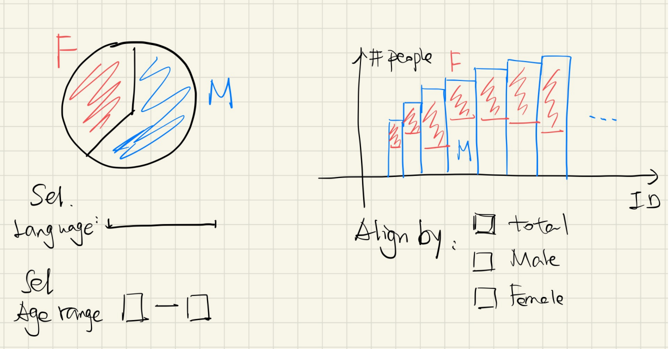


Fig. 2. Examples of distribution of the gender.

some languages with biased distribution, we should rule them out before our analysis. So the first visualization design we proposed is to show the distribution of participants.

Domain-specific Data:

The domain-specific data can be described in a four-tuple, i.e., (Language ID, Participant ID, Age, Gender), which is also the storage form of items in *spkr.txt*.

Abstracted Data:

In each item, the language ID and participant ID are natural numbers that can be considered as unordered classes. The age is an integer from 5 to 100, which can be divided into several intervals when visualizing the distribution. The gender can be considered as a boolean variable or a class with two possible values.

To visualize the distributions, we will first clean the data and make sure that all the remaining items are valid, i.e., gender is M or F, age is between 5 to 100. After that, we can count the items based on what information we plan to see in the figure.

3.2 Distribution of the gender

The visualization design should follow the requirements of the research task. We provide two examples in this section.

Example 1: accumulated distribution under specific conditions

Sometimes, the researchers might want to see the whether the gender distribution of specific groups of participants is uniform. Under this requirement, we chose to use a simple pie chart, with two colors encoding the ratio of male and female participants. Furthermore, we provide several options to control which participants are considered, as illustrated in the left panel of Figure 2. For example, one can narrow down the focus to all the valid participants speaking language 10 to 50, with ages range from 40 to 50, by setting parameters when calling this function.

Example 2: detailed distribution for different languages

Sometimes, the researchers are curious about the distribution (so as the total amount of participants) of each language they care about. Under this condition, the stacked bar chart can perfectly fulfill all the requirements. As illustrated in the right panel in Figure 2, the x-axis is the ID of different language and the y-axis represents the amount of participants. Each bar in this chart has two parts: blue-colored part of male participants and the red-colored part of female participants. The entire length of each stacked bar can represent the total amount of participants speaking this language. Besides the specific conditions we mentioned in example 1, one can also select how to align the stacked bars, e.g., align them by the number of male participants, by the number of female participants, by the ratio of male/female participants, by the total amount of participants, or by the language ID. With the help of this stacked bar chart, the user can rule out the language with too few participants or select the languages with unbiased male/female distributions.

3.3 Distribution of the age

Similar to the examples provided in the previous subsection, we can also use pie chart and stacked bar chart to analyze the age distribution, as illustrated in Figure 3. Before generating the figures, the users should first specify how many intervals of ages they want to observe, e.g., one can have 5 intervals: $(0,20], (20,40], (40,60], (60,80]$ and $(80,100]$. Note that the gap of different classes do not have to be the same. For the pie chart, the user can chose whether to consider only male (or female) participants or both. For the stacked bar chart, the user can chose whether align the bars by the total number of participants or by the language ID.

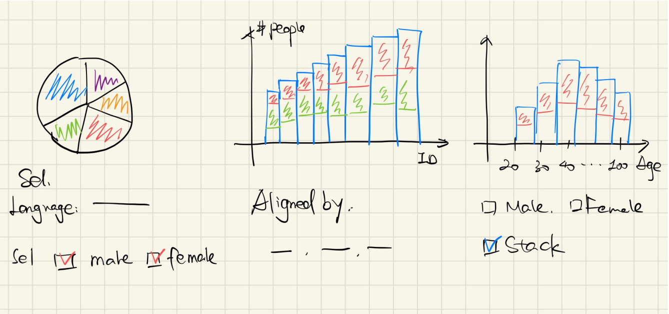


Fig. 3. Examples of distribution of the age.

Example 3: histogram of participants with different ages

In this task, the age attribute can be considered as the ordered class. Hence it is reasonable to use histogram to show the distribution, as illustrated in the right panel of Figure 3. The x-axis denotes the age intervals and the y-axis represents the number of participants. The user can also chose whether to show the distribution of male, female or all the participants. Note that the user can also use stacked bar histogram when observing the distribution of all the participants.

3.4 Discussions of vis idioms in this part

The tasks discussed in this part only focus on the distribution of the number of different participant types, hence the pie chart and the stacked bar chart are enough. Also, these two vis idioms have their pros and cons, the users can refer the following principles:

Pie chart is suitable when:

- you want to show the ratio of aggregate statistics;
- you want to convince the audience that one part occupies a larger ratio than another when their difference is nuance;
- you only want to put a few classes in the figure.

Stacked bar chart is suitable when:

- you want to see distribution of many classes (language or age) simultaneously;
- you want to order these classes to help you to select the satisfying class(es).

4 Visualization Design for Naming of a Single Language

The goal of WCS project is to provide insights on language evolution process. Usually, the researchers will first select some representative languages from the entire dataset (possibly, with the help of the methods we mentioned in the previous section), and then draw conclusions from the naming results. So the visualization design of the naming results for a single language is the most important part of our work.

4.1 Overview of this task

Domain-specific Data:

In the WCS project, the participants are asked to name some randomly selected chips on the Munsell color card using their mother language. Their responses are then mapped to abbreviations (with one or two letters) and stored in *foci-exp.txt*. The form of the dataset is illustrated in Table 1. There are five columns in this table:

- Language ID: denote which language this piece of recording belongs to, ranges from 1 to 110;
- Speaker ID: denote which speaker this piece of recording comes from;
- Focus response: sequential enumeration of focus responses, not important here;
- Term used: the abbreviation of the term used to describe the chip;
- Described chip: the (b,h) representation of the chip described in this recording.

Language	Speaker	Focus	Term	Chip
1	1	1	LF	A0
1	1	2	WK	D9
1	2	6	WK	D10
...	...	...	...	...

Table 1. Form of the data items in WCS dataset.

Abstract Data:

In each recording, the 'language ID' and 'speaker ID' are both non-ordered class. The 'term used' attribute is a string composed by 1 or 2 capital letters, which can also be considered as non-ordered class. The 'described chip' is the coordinate of the chip in the Munsell color card.

To visualize the naming results of one specific language, we will first collect all the items with that language ID and then make some counts. Based on the perspective we chose, i.e., the chip's view or term's view, we might have two different designs.

4.2 Design from chip's view

The chip's view design count on the spatial location of the chips on Munsell color card and put the corresponding terms on them, which matches our intuition well. To the best of our knowledge, most of the works on WCS dataset chose this design perspective, e.g., [3,4,6,8].

Chip ID	...	B01	B02	B03	...
Selected Term	...	GG	?	AG	...
Term(s)	...	GG(25)	-	AG(28)	...
	...	GA(3)	-	-	...

Table 2. Counts of term(s) for different chips, the number in the bracket behind term is the number of participant.

Before generating the figure, we should first count the term(s) used for each chip and fill Table 2. In the results collected from one language, it is possible for one chip to have multiple terms. Like for the chip B01, there are 25 participants use GG and 3 use GA. Under this condition, we choose the term used by the majority (i.e., GG for B01) as the selected term. Furthermore, it is also possible (actually, quite common) for one chip to have no records under it, like the B02 chip. Under this condition, we might have the following three methods to fill them:

- Allocate an unused term to ALL those chips;
- Allocate a term following the nearest chip in a row-first order;
- Allocate a term following the nearest chip in a column-first order.

The first method is suitable to use when we only draw chips with small amount of term types. As the example in Figure 4, we can allocate the same color to all the terms other than YA and SA (including those chips with no term collected) and treat them as the background. The figure is concise and easy to read. However, when we want to draw chips with multiple or all the term types, as illustrated in Figure 5, using the first method will harm the trend demonstrated by the figure (as these non-recorded chips are randomly distributed).

To solve this problem, we can allocate terms to these chips following the adjacent-similar principle. In the Muncell color card, the adjacent chips have similar hue and lightness. Hence it is reasonable to assume that the participants tend to use the same term with that naming the adjacent chips. Based on this fact, we might chose the second and the third method mentioned above. For the example in the upper panel in Figure 6, for a chip with no term, we will find the nearest recorded chip in x-axis to it and copy the term used by the recorded chip. The example in the bottom panel in Figure 6 do a similar thing following another axis. In these two examples, we can see the pattern of the figure is maintained (or even amplified). One thing to remember when using interpolation is that the three different methods mentioned above might influence the calculation of some quantitive metrics, e.g., topological similarity (we might discuss later). Hence we should compare such metrics under the same interpolation method.

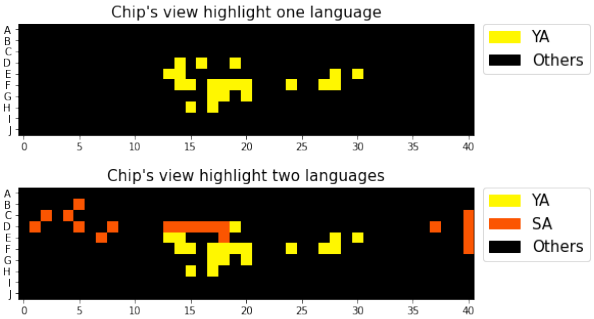


Fig. 4. Good examples of allocating unused term to the unrecorded chips when only several terms are highlighted.

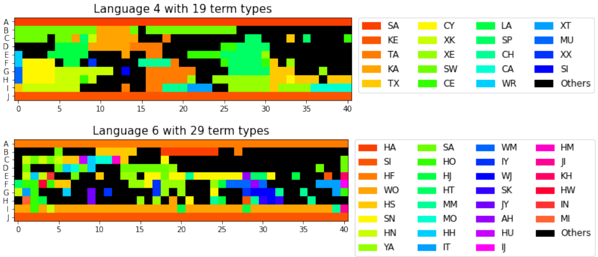


Fig. 5. Bad examples of allocating unused term to the unrecorded chips when all the terms are highlighted.

4.3 Design from term's view

Sometimes, the researchers may be curious about how many chips that each term can represent. At the same time, they might believe that interpolation of the unrecorded chips using adjacent principle can introduce bias. Under such a condition, a vis idiom from term's view is helpful. Compared with the chip's view we mentioned above, the biggest difference in term's view is that we no longer need to 'guess' the term for those unrecorded chips, which makes our figure unbiased to the real data.

Before drawing the figure, we also need to count the chip(s) represented by each term in the data. Similarly, we should fill Table 3. Note that in this table, each used term represent at least one chip. Besides, it is possible that one chip is represented by multiple terms.

Term	...	GG	GA	AG	...
Chip(s)	...	B01(25)	B04(28)	B03(22)	...
	...	B02(3)	-	-	...

Table 3. Counts of chip(s) for different terms, the number in the bracket behind term is the number of participant.

With the results in Table 2, we can visualize the data like Figure 7 does. In this vis idiom, we list all the possible terms used in this language, followed by all the chips it represents. To provide a better

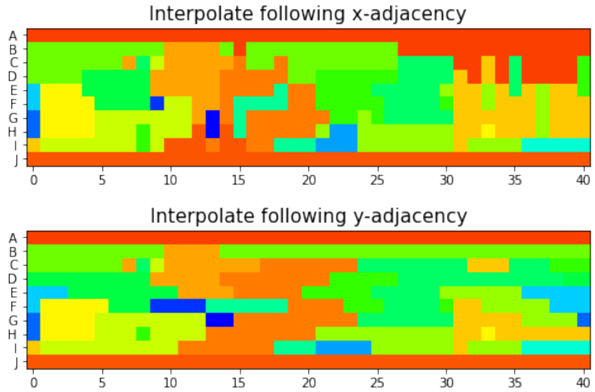


Fig. 6. Good examples of allocating unused term to the unrecorded chips following adjacency principles (**title of the figures are wrong**).

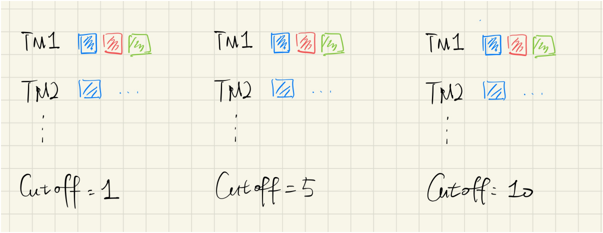


Fig. 7. Examples of term's view.

intuition of what color each chip represents, we use small squares with the color copied from Munsell card and align these squares to make the figure easier to read.

However, in some languages, there might be a lot of terms with only few recordings, which can make the figure too large and hard to read. To solve this problem, we design a parameter called 'cutoff number'. Specifically, the figure only show the chips that has enough records for each term. In the example in Table 3, when the cutoff number is set to 5, the chip B02, which only has 3 records, will not appear behind the term GG. The cutoff number should be carefully selected to make the figure concise and informative: a too small value will make the figure containing too much noisy information (like the upper panel in Figure 7) while a too large value will lose important information (like the bottom panel in Figure 7).

4.4 Discussions of vis idioms in this part

The task of designing visualization for naming of a single language is more complex compared with the task in the previous section: there are so many features to think about.

Spatial location on Muncell:

In this task, each chip has a unique location on the Muncell card and the spatially adjacent chips have similar hue and lightness. Such an adjacent-similar fact can help us to make interpolation for the unseen records. Plus, drawing items on top of the grid of Muncell card (like example in Figure 4, 5 and 6) makes it easy to find patterns on the figure.

Color coding:

Another important problem to consider is the color coding method. Generally, there are two ways: following Munsell or not. The first choice matches our intuition well (like examples in Figure 5[**figure not ready yet**]), but the similarity among the chips with the same term (so as the distinction among the chips with different terms) is hard to observe. The second choice can avoid this problem by setting high-contrast colors to different chips (like example in Figure 6) or just consider non-highlighted parts as background and allocate black color to them (like examples in Figure 5). Such a method can amplify the distinctness between different region, but lose the information of the true color of the chips.

Some lost information:

In all the methods we discussed above, we will first count participants that map a specific term to a chip (or vice versa). For those terms with multiple chips (or chips with multiple terms), we usually ignore the mappings with fewer recordings, which means that the information of divergence of the mapping (i.e., the language) is missing. To solve this problem, we tend to prose a quantitive metric called inconsistency of one language in the next section.

In summary, the users should select an appropriate visualization design first, then carefully select and align all the parts in the figure to make it readable.

5 Visualization Design for Quantitive measurements

5.1 Overview of this task

In many of the work on WCS data, researchers will calculate some quantitive measurements for different attributes from the raw data of WCS and observe the influence of these metrics. For example, [1] proposes a metric called TRE to measure the structureness of one language, [8] analyzes the mutual information between chips and terms, [2] applies topological similarity to measure the compositionality of a language. What is the correlation among these quantitive measurements then becomes an interesting topic to explore. As this work focus on visualization design rather than evolutionary linguistics, we will not go deeper into these metrics. Rather, we will introduce some vis idioms based on three fundamental and widely applied metrics to show how our visualization toolkit can help. The metrics (can be considered as abstract data) we discussed are:

- Topological similarity (topsim for short), proposed in [2], a widely used metric for the compositionality of one language. The range is from 0 to 1;
- Inconsistency (incons for short), a metric to evaluate the extent of divergency of mappings between terms and chips. As a ratio metric (number of diverged mappings to the total mappings), its range is 0 to 1;
- Term types, a metric to evaluate how many different terms are applied in describing call chips on Munsell card. It is a natural number.

5.2 Distribution of one metric

The task of visualizing the distribution of a single quantitive value is similar to the problem we discussed in section 3, i.e., visualizing the distribution of participants. However, the researchers usually want to see how the demonstrated metric influence the language. To fulfill this requirement, we make the following design by combining the idioms introduced in both section 3 and 4.

Example X: influence of toplogicial similarity

The main body of the figure is a bar chart: the x-axis represents different languages and the y-axis represents the topsim value. To make the figure more readable, we align all the bars by the values of topsim. On top of this bar chart, we use two lines (parallel with the x-axis) to show the mean and the median of all the topsim values. Furthermore, to provide a general view of how topsim influence the pattern in one language, we select 3 languages (with low/medium/high values of topsim) and generate 2 subfigures for each of them.

To better demonstrate the compositionality, we apply the chip's view design in section 4 and make some modifications to make the figure easier to read. As a highly compositional language (i.e., language with high topsim) tend to use each of its letter to represent different attributes (i.e., hue and lightness in WCS data) and the vocabularies it applied would be consistent, it is reasonable to use two subfigures for each position of the phrase respectively. Specifically, in the first subfigure, we merge all the terms starting with the same letter to one large group (e.g., GA and GG can be merged to G*). In the second subfigure, we merge all the temrs ending with the same letter (e.g., GA and SA can

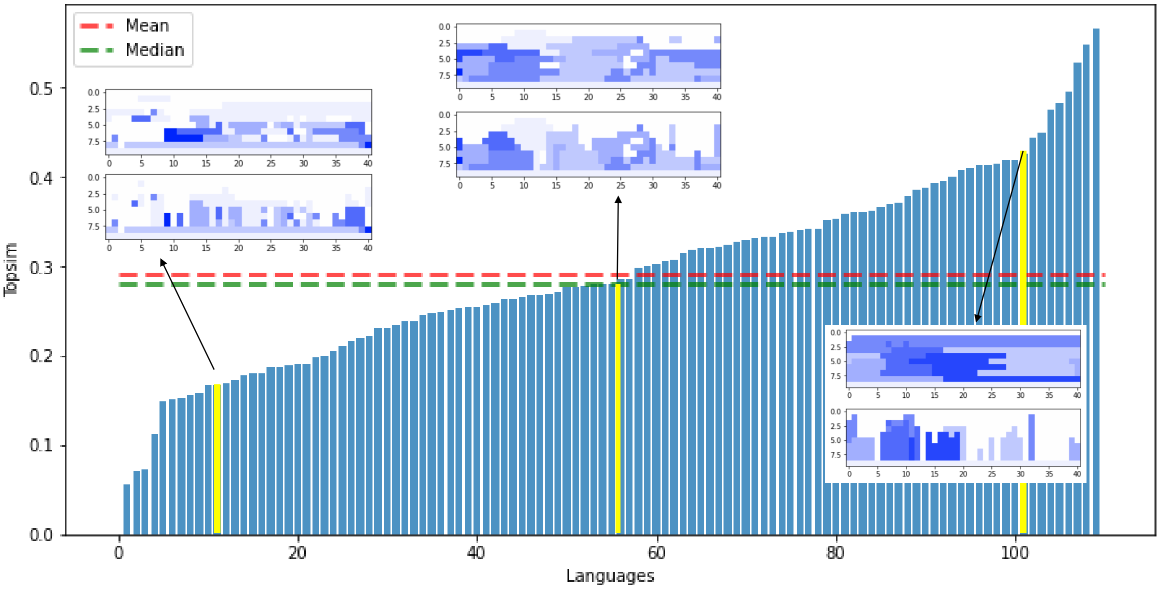


Fig. 8. Joint plot of topsim's distribution, examples of high/medium/low topsim languages.

be merged to *A). Such a design can significantly reduce the number of groups in the subfigure, and hence make it easier to read. Then we draw two figures for each selected language, and put them beside their bars. From the whole visualization, the readers can clearly see the distribution of topsim over all the recorded languages, so as the mean and medium value. Plus, they can also get an intuition from the 3 selected language: the terms of the language with high topsim is better aligned from chips' view visualization.

Such a design can also be applied to show the influence of inconsistency or term types.

5.3 Correlation among many metrics

The previous subsection discusses how to show the results of one metric. Sometimes, the researchers may be curious about the correlation among different metric. Hence in this part, we propose several vis idioms to show this information.

Example X: correlation among multiple metrics

To see the correlation among two attributes, an appropriate vis idiom is the combination of scatter plot and line chart for linear regression (with the shadow region representing the 90% confidence interval), as illustrated in Figure 9. The marginal distributions along the two axis can also provide useful information.

Furthermore, if we want to observe the correlations among three attributes in one figure, we might use the color or size channel of spots to encode one attribute. For example, in Figure 10, the x and y axis represent two attributes. The color channel encodes another attribute using different level of hue. In this design, it is only possible to draw regression line between the value encoded by two axises. Hence it is better to allocate the color channel to the unimportant attributes.

5.4 Discussions of vis idioms in this part

To be done ...

6 Conclusion

To be done ...

7 Milestone

Nov. 18 to Nov. 25 (One week)

Accomplish writing section 5.

Further specify the what/why/how principles for each vis task mentioned in the paper.

Accomplish the code for generating Figure 2,3. Based on the knowledge learned from VAD, polish section 3.

Consider how to implement color coding using the corresponding Munsell's color (at specific location).

Nov. 26 to Dec. 2 (One week)

Accomplish the code for generating Figure 7. Polish section 4.

Dec. 2 to Dec. 5 (3 days)

Accomplish introduction, abstract, conclusion parts.

Dec. 5 to Dec. 9 (4 days)

Pack-up the codes, polish the paper.

References

[1] J. Andreas. Measuring compositionality in representation learning. *arXiv preprint arXiv:1902.07181*, 2019.

[2] H. Brighton and S. Kirby. Understanding linguistic evolution by visualizing the emergence of topographic mappings. *Artificial life*, 12(2):229–242, 2006.

[3] E. Gibson, R. Futrell, J. Jara-Ettinger, K. Mahowald, L. Bergen, S. Ratnasingam, M. Gibson, S. T. Piantadosi, and B. R. Conway. Color naming across languages reflects color use. *Proceedings of the National Academy of Sciences*, 114(40):10785–10790, 2017.

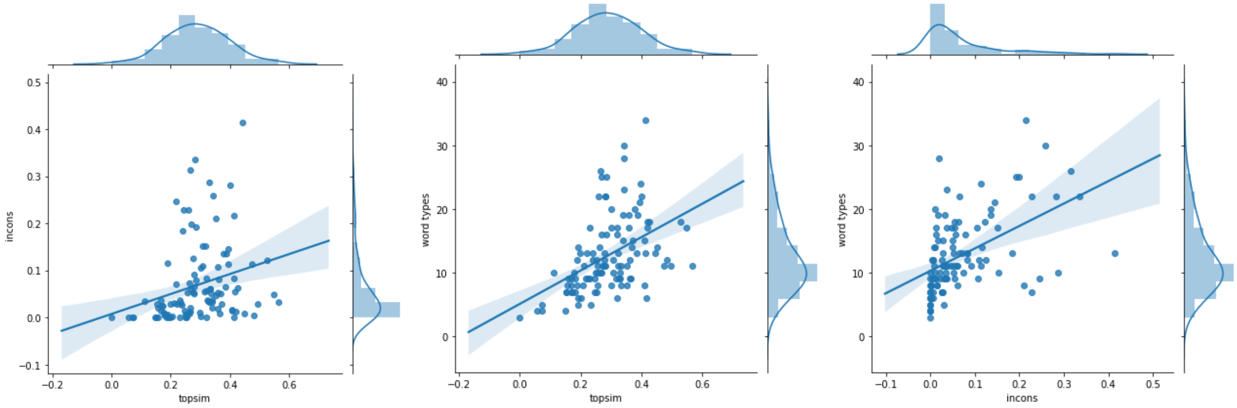
[4] D. T. Lindsey and A. M. Brown. World color survey color naming reveals universal motifs and their within-language diversity. *Proceedings of the National Academy of Sciences*, 106(47):19785–19790, 2009.

[5] T. Munzner. *Visualization analysis and design*. CRC press, 2014.

[6] T. Regier, P. Kay, and N. Khetarpal. Color naming reflects optimal partitions of color space. *Proceedings of the National Academy of Sciences*, 104(4):1436–1441, 2007.

[7] P. K. Richard Cook and T. Regier. Wcs data archives.

[8] N. Zaslavsky, C. Kemp, T. Regier, and N. Tishby. Efficient compression in color naming and its evolution. *Proceedings of the National Academy of Sciences*, 115(31):7937–7942, 2018.

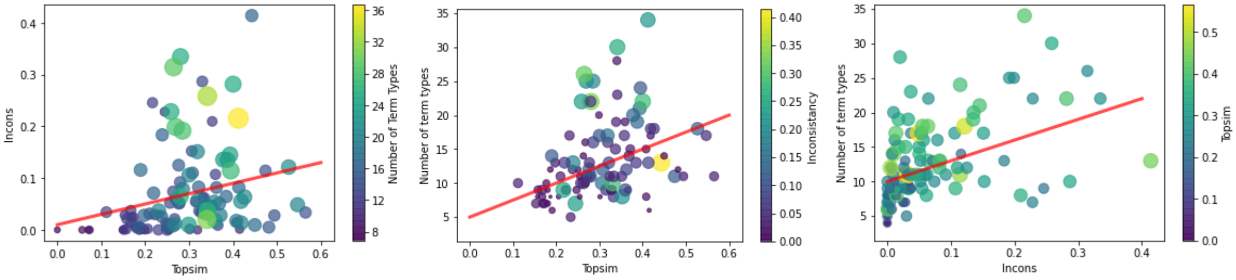


(a) Topsim v.s. incons.

(b) Topsim v.s. term types.

(c) Incons v.s. incons.

Fig. 9. Joint plot of scatter plot, linear regression and distribution.



(a) Topsim v.s. incons.

(b) Topsim v.s. term types.

(c) Incons v.s. incons.

Fig. 10. Using size and color channel to encode another attribute.