Comparing Apples and Oranges: Taxonomy and Design of Pairwise Comparisons within Tabular Data

Po-Ming Law
Georgia Institute of Technology
Atlanta, Georgia
pmlaw@gatech.edu

Subhajit Das
Georgia Institute of Technology
Atlanta, Georgia
das@gatech.edu

Rahul C. Basole
Georgia Institute of Technology
Atlanta, Georgia
basole@gatech.edu

ABSTRACT

Asking pairwise comparison questions is common. Yet, we often find ourselves comparing apples and oranges — the two entities of interest are not readily comparable. To understand how technologies can extend our capabilities to conduct pairwise comparisons during data analysis, we analyzed pairwise comparison questions collected from crowd workers and propose a taxonomy of pairwise comparisons. We demonstrate how the taxonomy can be adopted by incorporating pairwise comparison capabilities into Duo, a spreadsheet application that supports comparing two groups of records in a data table. Duo decomposes a pairwise comparison question into rules and showcases sloppy rules, a query technique for specifying pairwise comparisons. We conducted a user study comparing sloppy rules and natural language. The findings suggest that for easier pairwise comparison tasks, the two techniques are comparable in efficiency and preference and that for more difficult pairwise comparison tasks, sloppy rules allow faster specification and are more preferable.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**;

KEYWORDS

Comparison; spreadsheet; query specification; natural language; data analysis

ACM Reference Format:
Po-Ming Law, Subhajit Das, and Rahul C. Basole. 2019. Comparing Apples and Oranges . In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland UK*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3290605.3300409

CHI 2019, May 4–9, 2019, Glasgow, Scotland UK

ACM ISBN 978-1-4503-5970-2/19/05…$15.00
https://doi.org/10.1145/3290605.3300409

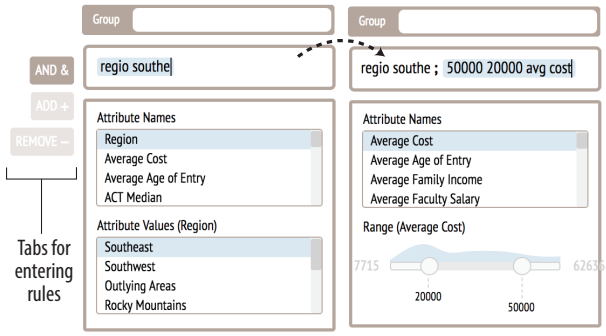

Figure 1: A user is entering sloppy rules to specify a group of colleges where `Region`=*Southeast* AND *20000*⩽`Average Cost`⩽*50000*. The sloppy rule menu has AND, ADD, and REMOVE tabs for entering base, inclusion, and exclusion rules respectively.

1 INTRODUCTION

Pairwise comparisons penetrate many aspects of our lives. Consider a colleague who shares her experience of visiting Stockholm by drawing an analogy between the busiest areas there and New York. The comparison may conjure up an image of the big city bustle in Stockholm even you are not familiar with the city. We all may recall moments when we compare ourselves with our peers to strive for improvement. Marathoners, for example, compare themselves to similar runners on finishing time to gain accurate self-assessments.

More broadly, pairwise comparisons have fundamental influence on societal development. In a world where resources are limited, decision makers often need to understand trade-offs between two options to make a well-informed decision about which one to choose. Through pairwise comparisons, important questions can be answered, our inquisitive nature is satisfied, and human knowledge is advanced. Behavioural scientists, for instance, are studying the suicide crisis in the US Army by comparing military personnel to the general public [32]. Such research can provide better understanding of humans and save millions of lives.

The comparisons we just described share a commonality: the entities being compared are apples and oranges: Stockholm and New York, albeit similarly prosperous, are distinct in many ways; a single marathoner is clearly different from the thousands of others; military service members and the general public are substantially different in their geographical distributions and populations. While many statistical techniques (e.g., Tukey's HSD test [39]) exist to facilitate pairwise comparisons, the learning curve is prohibitively high for laypeople. Interactive systems reduce the barriers to pairwise comparisons. Yet, most are limited to comparing particular kinds of objects (e.g., [21, 26]) and have restricted comparison capabilities (e.g., [14, 19]).

In this paper, we inquire into how technology can extend our cognitive and analytic capabilities so that comparing apples and oranges is more surmountable. First, we solicited pairwise comparison questions from 398 crowd workers. We report a qualitative analysis of the questions that culminates in a taxonomy of pairwise comparisons. The taxonomy uncovers new possibilities for interactive data analysis systems by showing what pairwise comparisons they could support.

Second, grounded in the taxonomy, we demonstrate how the most widely used data analysis tool — spreadsheets — can be augmented by incorporating pairwise comparison capabilities into a spreadsheet application called Duo. Duo considers a pairwise comparison question as a set of rules: base rules, inclusion rules, exclusion rules and attributes. To help users articulate complex comparisons, we explored sloppy rules, a query technique for specifying pairwise comparison rules. Inspired by research in linguistic commands [16, 23, 25, 33], sloppy rules support variations, forgive spelling errors, are order-independent, and offer rich visual feedback.

Finally, we conducted a within-subject study during which 16 participants used both the sloppy rule interface and a natural language interface to perform pairwise comparison tasks of varying levels of difficulty. The results indicate that the two interfaces are comparable for simple pairwise comparisons but that the sloppy rule interface is more efficient and preferable for more difficult pairwise comparisons. Based on the study results, we discuss considerations in designing query interfaces for pairwise comparisons.

2 RELATED WORK

Besides a prior framework of comparison [13], we surveyed the literature on functions of pairwise comparisons when developing our taxonomy. We built on other pairwise comparison tools, and linguistic commands when designing Duo.

Functions of Pairwise Comparison

Pairwise comparison serves as an important mental tool in four motivations and objectives: analogical reasoning, social comparison, decision making, and knowledge discovery.

Analogical reasoning is a process of understanding a novel situation in terms of familiar ones [11]. It involves a base (a familiar entity) and a target (an unfamiliar entity) [12]. Through a mapping process, we align the representations of the base and the target, and project inference from the base to the target [12]. Pairwise comparison is an essential tool during the process of alignment.

Social comparison theory [8] states that we are driven to determine our own worth by comparing ourselves against people who are better or worse off than us. Interactive systems have been developed to help us compare ourselves with similar people to predict our futures and devise actions [4, 5].

Pairwise comparisons shape many of our decisions (e.g., [38, 40]). In operations research, multi-criteria analysis techniques, such as the analytic hierarchy process, use pairwise comparison for decision making [36].

We also conduct pairwise comparisons for pure pursuit of knowledge that may not have an immediate purpose, and is often driven by our curiosity [15] and intrinsic motivation [35]. Knowledge distilled from pairwise comparisons is often used for deriving action plans (e.g., [34]).

Based on the above pairwise comparison functions, we designed scenarios to collect pairwise comparison questions from crowd workers. The analysis of these questions provides the basis for a taxonomy of pairwise comparisons.

Pairwise Comparison Tools for Tabular Data

Statistical tools such as R [9] are used by researchers to conduct pairwise comparisons (e.g., Tukey's HSD test [39]). However, laypeople can struggle to use these techniques due to a lack of statistics background. One of our goals is to empower a non-expert audience to conduct pairwise comparisons on tabular data. We do so by bringing pairwise comparison capabilities to the widely-used spreadsheets.

Research in information visualization has been conducted to explore different ways of comparing tabular data. TACO [24] visualizes how a table changes over time. Kehrer et al. [18] proposed a model that formalizes comparison of cells within a small-multiple display. Duet [19] employs the minimal specification technique to enable pairwise comparisons of records in a data table. Our work furthers this line of research by offering a systematic crowdsourced study of what pairwise comparisons are possible and proposing the sloppy rule technique for specifying pairwise comparisons.

Some online tools seek to support head-to-head comparisons of certain objects. For instance, US News allows web users to choose a pair of colleges, and juxtaposes them for comparison [21]. Many spreadsheet applications are also endowed with some pairwise comparison capabilities. Excel, for example, provides pivot tables for data grouping and aggregation [22]. The "Explore" button in Google Sheets allows users to enter simple comparison questions using natural

language [14]. However, these tools do not support some commonly-performed pairwise comparisons revealed by our taxonomy (e.g., one-to-multiple comparisons).

Linguistic Commands

When it comes to asking pairwise comparison questions, we would immediately think of using natural language due to its intuitiveness. For in-depth pairwise comparisons, however, questions can become verbose. Prior work showed that natural language-only interfaces are inferior to multi-modal interfaces due to high cognitive load in articulating queries [28–30]. Based on their findings, we hypothesize that high cognitive load could make it difficult to articulate long pairwise comparison queries using natural language.

To deal with complex queries, a reversal to command lines was proposed. Linguistic commands [16, 25, 33] are queries that exhibit touches of natural language and are tolerant of variations. Inky [23] and Chickenfoot [1, 20] showcased the power of linguistic commands. Inky adopts sloppy syntax to offer quick access to common web tasks such as reserving conference rooms. Different from conventional command lines, sloppy syntax was designed to be order-independent and robust, shielding users from memorizing syntax. For example, *"reserve Room123 3pm"* and *"3pm Room123 reserve"* will produce the same interpretation. Ubiquity [6] and Quicksilver [31] similarly provide linguistic commands for retrieving information and requesting actions. To tackle complex pairwise comparison queries, we explore sloppy rules, a robust linguistic commands for specifying pairwise comparisons.

3 A TAXONOMY OF PAIRWISE COMPARISONS

Informed by the functions of pairwise comparisons in the literature, we designed common daily life scenarios in which pairwise comparisons are instrumental. Using the scenarios, we solicited pairwise comparison questions through Amazon Mechanical Turk (AMT) and coded the resulting corpus of questions. Grounded in the analysis, we propose a taxonomy of pairwise comparisons. This taxonomy provides insights into what pairwise comparison capability interactive data analysis systems could support.

Method

To understand the kinds of pairwise comparison questions people might ask, we considered gathering questions from data analysts through interviews. However, this approach limits the number of people from which the data are collected. The questions collected through interviews will also be restricted in scope and highly specific to a few domains.

To ensure diversity in the questions, we collected pairwise comparison questions from workers on AMT. Based on the four types of pairwise comparison functions (i.e. analogical reasoning, social comparison, decision making, and knowledge discovery), we designed eight daily life scenarios (two for each type). To make these scenarios more realistic, we drew inspirations from scenarios we observed from popular sites (e.g., reddit.com). The eight scenarios are included as supplemental material[1]. As an example, here is one of the two scenarios that involve social comparison:

Imagine that you plan to run a marathon next year and want to finish the marathon in 3 hours. You are curious about how your fitness level compares with those who were able to finish the marathon in 3 hours so you obtain a dataset of the runners who participated in last year's marathon. Your data contains 200 runners who finished last year's marathon in 3 hours. You know the runners' body mass index (BMI), resting heart rate, diet, how many miles they ran daily for training, how long they had trained before the race, and so on. If you want to compare your fitness level with these runners, what question would you ask?

Given a scenario, workers were required to provide a question they would ask. We limited the tasks to workers who had completed at least 50 HITs and had an acceptance rate of 95% or above. The workers were compensated $0.19 for each question they provide (hourly wage = $7.25) to align with the Ethical Guidelines for AMT Research [37].

Analysis. In total, we collected 1150 questions from 398 unique workers. The first author open-coded the collected questions before paring down to four dimensions of codes and developing the codebook. Two researchers independently coded the entire corpus along the four dimensions. During the coding process, the coders independently coded some questions, discussed inconsistencies, refined the definitions of codes, and recoded some questions independently based on the new definitions. The analysis codebook, the questions, and the detailed statistics are included as supplemental material.

The first dimension has three categories: *irrelevant*, *implicit* and *explicit* (Cohen's κ = .72). We coded a question as *explicit* if it is an obvious pairwise comparison question, and as *irrelevant* if it is not a pairwise comparison. We introduce a category, *implicit*, to deal with questions that seems to be a step in a pairwise comparison task but do not explicitly compare two entities. For the above scenario, an example is *"What is the average number of hours they slept per day?"* While this question does not seem to involve comparison, knowing other runners' sleep duration is a necessary step before one can compare herself to other runners on sleep duration.

[1]Supplemental materials including data, working prototypes, demo videos, and all materials for the user study in Section 5 can be found here: **https://github.com/duospreadsheet/supplemental**

To avoid over-interpreting people's intent, we omitted the questions coded as *irrelevant* and *implicit*. 549 pairwise comparison questions coded as *explicit* remained. These questions were further coded along three dimensions: repetition, group, and attribute. The repetition dimension has two categories: *single*, and *repeated* (κ = .67); the group dimension has three: *one to one*, *one to multiple*, and *multiple to multiple* (κ = .76); the attribute dimension has two: *attribute present*, and *attribute absent* (κ = .86). These three dimensions form the skeleton of our taxonomy of pairwise comparisons.

Limitations. Our study is limited in that the frequency of a category does not reflect the frequency that a particular kind of pairwise comparison appears in our daily lives or in data analysis. This is because our study relied on designed scenarios to elicit pairwise comparison questions from crowd workers rather than observing pairwise comparison questions in the wild which is arguably impossible to do at scale. This implies that some types of pairwise comparison questions revealed by our analysis might be observed less frequently than people would actually ask about their data.

Nevertheless, our study provides evidence for the *presence* of various pairwise comparison questions. The wide coverage our taxonomy serves as the foundation for tools that aim to support a wide range of questions.

The Taxonomy

Underpinning our taxonomy are three components of pairwise comparisons: repetition, group and attribute, each corresponding to a dimension of codes. The three dimensions form 12 types of pairwise comparisons (2 categories from the repetition dimension $\times$ 3 from group $\times$ 2 from attribute).

Repetition. Conventionally, pairwise comparisons have the form "compare A to B" (category = *single*). Yet, we also identified questions in which multiple pairwise comparisons are embedded. These pairwise comparison questions share the form "compare A to B, C, D, etc." (category = *repeated*). In such questions, people repeatedly perform pairwise comparisons using A as an anchoring point: A is compared to B, to C, then to D and so on. Here are two examples:

Q1. *Which US city is Rovaniemi most similar to?*

Q2. *Is Rovaniemi spread out like Los Angeles or is it more compact like New York?*

Both questions follow the "compare A to B, C, D, etc." pattern: **Q1** compares Rovaniemi (A) to different US cities (B, C, D, etc.); **Q2** compares Rovaniemi (A) to Los Angeles (B) and to New York (C). People often ask this type of questions when they are not familiar with entity A and want to gain an understanding of it by comparing entity A to some familiar entities. For example, in **Q2**, the asker wants to get a sense of how sprawling is the unfamiliar Rovaniemi by comparing it to the more familiar Los Angeles and New York. An answer to a repeated comparison can be an entity that is the most similar to or the most different from entity A among entities B, C, D, etc. An answer to **Q1**, for example, is a US city that shares many similar characteristics with Rovaniemi.

Group. A pairwise comparison involves two groups for comparison. The two groups can both be single-object groups (category = *one to one*). For example, *"What are the major differences between Mars and Earth"* involves comparing a single-object group to another single-object group (Mars vs Earth). Sometimes, a pairwise comparison question consists of a single-object group and a group that contains multiple objects (category = *one to multiple*). This is common in social comparison in which we compare ourselves to our peers (e.g., *"Is the amount I run every day for training the same or higher than the average of the other runners"*). Comparing a multiple-object group to another multiple-object group can also be found in our corpus (category = *multiple to multiple*). An example is *"In the three largest US cities, is there a higher or lower crime rate than the three largest UK cities?"* The asker attempts to compare a group of three UK cities to another group of three US cities.

Attribute. It is common to ask a pairwise comparison question by specifying an attribute of interest (category = *attribute present*). For instance, when people ask, *"What are the temperature differences between Mars and Earth"*, temperatures of the two planets are being compared. We also observed questions where people do not state an attribute (category = *attribute absent*). Examples are *"What are some similarities and obvious differences between monkeys and humans"* and *"How much difference is it between my phone and the Galaxy Note 5"*. People often ask these questions when they want to understand similarities and differences of two entities broadly and do not have a good sense of what attributes are important.

Gleicher proposed a related framework that comprises considerations for designing comparison tools [13]. Using his terminology, our taxonomy serves to *identify the comparative elements* in pairwise comparisons: the group dimension depicts the characteristics of the comparison *targets* whereas the attribute and repetition dimensions provide details about the *actions* people do on these comparison *targets*.

4 DUO: A SPREADSHEET APPLICATION FOR PAIRWISE COMPARISONS

While the previous section elucidates the what of pairwise comparisons, this section describes the how: how can users specify pairwise comparison questions and how should these questions be answered? We demonstrate how techniques for conducting pairwise comparisons can be embodied in

spreadsheets, arguably the most widely-used data analysis application for non-experts. To this aim, we developed a spreadsheet application called Duo, which supports the 12 types of pairwise comparisons on wide-format tables.

To explain various techniques, we use a US college dataset [27] as a running example. This dataset contains 1214 US colleges, each consisting of 4 categorical attributes (e.g., `Region`) and 20 numerical attributes (e.g., `Admission Rate`).

Specifying Pairwise Comparison Questions

Natural language is a compelling technique for specifying pairwise comparison questions. Yet, vicissitudes of natural language parsing can hamper user experience. Leveraging the insight that pairwise comparison questions can be decomposed in a standard way, we introduce a query technique for pairwise comparisons without natural language.

Unpacking Groups in Pairwise Comparisons. Groups are core components in a pairwise comparison question. They can be formulated using the logical operators **AND**, **OR**, and **NOT**.

An example question that involves **AND** is *"Of cities with greater than 100,000 residence, does the UK or US have more crime?"* In the example, cities that have more than 100,000 people **AND** in the UK are compared to cities that have more than 100,000 people **AND** in the US. **OR** can be used for composing custom groups. For instance, people may want to compare the Eastern and the Western worlds and have a city dataset. Each city may have a `Continent` attribute but not an attribute that indicates whether it is in the Eastern or the Western hemisphere. Using **OR**, people can compare cities in Asia (the Eastern world) to cities in Europe **OR** North America **OR** Australia (the Western world). Removing objects from a group before conducting pairwise comparisons can be achieved using **NOT**. This is common in research studies in which researchers drop outliers before hypothesis testing.

While groups can be fully described by these logical operators, non-experts can find it challenging to understand concepts such as operator precedence. We also observed that **AND** is more common than **NOT** and **OR** for defining groups. To simplify the mental model in defining groups and prioritize **AND**, we propose base, inclusion and exclusion rules for group definition that are described in the next section.

Decomposing Pairwise Comparisons into Rules. Rules constitute the skeleton of a pairwise comparison. There are two kinds of rules: **1)** rules for group definition and **2)** attributes under comparison. For example, *"compare the completion rates of the colleges in the Southeastern US with a low admission rate to those in New England with a low admission rate"* can be restated as *"compare the colleges where* `Region`*=Southeast and* `Admission Rate`*<0.2 to colleges where* `Region`*=New England and* `Admission Rate`*<0.2 on* `Completion Rate`*"*. The query has five rules: `Region`*=Southeast* and `Admission Rate`*<0.2* are rules for defining the first group; `Region`*=New England* and `Admission Rate`*<0.2* are rules for defining the second group; `Completion Rate` is the attribute under comparison.

Table 1: Correspondence between AND, OR and NOT logical operators, and base, inclusion and exclusion rules.

	Group	Rules
(a)	Colleges where `Region`=*New England* **AND** `Admission Rate`⩽*0.2*	`Region`=*New England* (base) `Admission Rate`⩽*0.2* (**base**)
(b)	Colleges where `Locale`=*Fringe Rural* **OR** `Locale`=*Distant Rural* **OR** `Locale`=*Remote Rural*	`Locale`=*Fringe Rural* (base) `Locale`=*Distant Rural* (**inclusion**) `Locale`=*Remote Rural* (**inclusion**)
(c)	Colleges where Region=*New England* and **NOT** `Name`=*Harvard College*	Region=*New England* (base) `Name`=*Harvard College* (**exclusion**)

Duo further decomposes rules for group definition into base, inclusion, and exclusion rules, which roughly correspond to **AND**, **OR**, and **NOT** respectively (see Table 1 a-c). Base rules describe the basic characteristics of a group. For example, *"colleges in New England with admission rate ⩽ 0.2"* consists of the colleges where `Region`*=New England* (base rule) and `Admission Rate`⩽*0.2* (base rule). Inclusion rules modify a group by adding objects to the group. *"Fringe, distant and remote rural colleges"* contains the colleges where `Locale`*=Fringe Rural* (base rule), together with the colleges where `Locale`*=Distant Rural* (inclusion rule) and those where `Locale`*=Remote Rural* (inclusion rule). Finally, exclusion rules remove objects from a group. *"New England colleges except Harvard"*, for instance, comprises the colleges where `Region`*=New England* (base rule) but not `Name`*=Harvard College* (exclusion rule). In freeing users from learning concepts such as operator precedence, expressivity is sacrificed as some complex logical expressions cannot be directly converted to the three kinds of rules.

Using Sloppy Rules for Rule Specification. In an informal usability study, we asked participants to perform pairwise comparison tasks using both a graphical user interface (GUI) and a natural language interface (NLUI). The participants specified pairwise comparison rules by point and click using the GUI and entered pairwise comparison sentences using the NLUI. They commented that it was harder to use the NLUI to *"think about"* and *"articulate"* long queries. While they found the GUI easier for articulating long queries, the interface complexity counteracted with its ease of articulation. Our pursuit of a technique with both ease of articulation and simplicity in operation resulted in the sloppy rules.

There are two group shelves and an attribute shelf in the sloppy rule interface (Fig. 2a). To define a group being compared, users click on a group shelf and a menu is shown

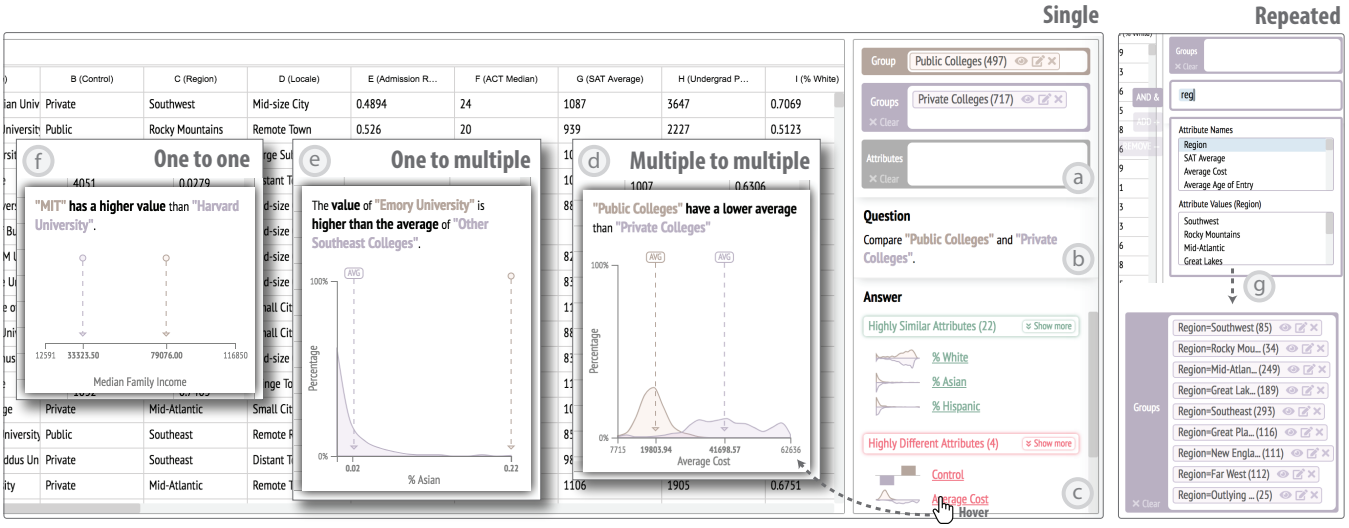

Figure 2: (a) Duo has three shelves for specifying pairwise comparison rules, and (b) displays the comparisons being performed in text. (c) For comparisons with the form *"compare A to B"*, it computes the similar and different attributes between the two groups. (d–f) When users hover over an attribute, the distributions are visualized. (g) When users partially specify an attribute but not a value using the sloppy rule menu, multiples groups are created to facilitate *repeated* pairwise comparison.

(Fig. 1). There are three tabs in the menu. Users enter the base rules in the **AND** tab, the inclusion rules in the **ADD** tab and the exclusion rules in the **REMOVE** tab. Multiple rules of the same kind are separated by semi-colons. Users can navigate through the tab by pressing the up and down arrow keys and confirm rule specification by pressing enter. To define attributes for comparison, users click on the attribute shelf and enter attributes separated by semi-colons.

Rules entered between semi-colons can be highly sloppy. Instead of complete rules (e.g., `Region`=*Southeast* ; Completion Rate$\geqslant$*0.8*), users can enter partial rules (e.g., `rg`=*s-east* ; `complete`$\geqslant$*0.8*) to define a group. This saves users the frustration especially when rules are long (e.g., `Name`=*Georgia Institute of Technology*). Following the sloppy syntax [23], four characteristics endow sloppy rules with their sloppiness and robustness:

Support for variations. Duo recognizes multiple variations of the same rules. For categorical attribute-value pairs such as `Region`=*Southeast*, Duo accepts `Region`=*Southeast*, `Region` is *Southeast*, and `Region` *Southeast*. For numerical attribute-value pairs such as `Completion Rate`$\geqslant$*0.8*, Duo interprets `Completion Rate` greater than or equal to *0.8*, and `Completion Rate` *0.8* or above as the same rule.

Tolerance for spelling errors. Under the hood, Duo computes the edit distances between a user input and attribute names and values. Hence, `rg`=*s-east* and `reg`=*southe* are both interpreted as the rule `Region`=*Southeast*.

Order independence. The order of attribute names and values does not matter. As an example for numerical attributes, *20000* `Average Cost` *50000*, `Average Cost` *50000 20000*, and *50000* `Average Cost` *20000* are all recognised as the rule *20000*$\leqslant$`Average Cost`$\leqslant$*50000*. For categorical attributes, users can directly enter the attribute value (e.g., *Public*) rather than specifying the complete rule (e.g., `Control`=*Public*).

Rich visual feedback. An overarching challenge in making sloppy rules usable lies in reducing the gulf of evaluation: how do users know if Duo successfully interprets their inputs when they enter a partial rule (e.g., `con` is *pri*) in place of a complete rule (e.g., `Control`=*Private*)? Without any feedback, users are more likely to resort to complete rules to ensure the system understands their inputs correctly, nullifying the benefits of sloppy rules. To reduce the gulf of evaluation, Duo provides rich visual feedback. While users are typing, Duo highlights its interpretation of what attributes and values are being entered (Fig. 1 left). For numerical attributes, Duo shows a slider with an embedded density plot (Fig. 1 right) and adjusts the slider range based on the current input.

Duo further supports multiple ways for entering a rule. Aside from typing a rule in the textbox and clicking on attribute names and values from the sloppy rule menu, Duo takes advantage of the data table to reduce the effort in entering rules. When users select a group shelf to open the sloppy rule menu, and click on a cell (e.g., a cell with the value *Southeast*), Duo adds the corresponding attribute-value pair (e.g.,

`Region=`*Southeast*) to the input box; when users select an attribute shelf and then a column header (e.g., a column header with the attribute `Average Cost`), Duo adds the attribute (e.g., `Average Cost`) to the input box.

After entering the rules that define a group, users press enter to confirm the rules. A tag that represents the created group is added to a group shelf. Users can click on to edit the rules. They can also click on to view what records are included in or excluded from the group. This allows users to double-check if the intended group is correctly created.

Specifying Comparisons Using Sloppy Rule Interface. Duo supports both forms of pairwise comparisons: "compare A to B" (*single*) and "compare A to B, C, D, etc" (*repeated*).

An example of "compare A to B" is *"compare colleges in the Southeast to colleges in New England"*. To specify the comparison, users first add a group with `Region=`*Southeast* as a base rule to the top group shelf and then add a group with `Region=`*New England* as a base rule to the bottom group shelf.

For comparisons with the form "compare A to B, C, D, etc", users create group A (the anchoring point) using the top group shelf and adds groups B, C, D, etc to the bottom group shelf. Duo allows partial specification to facilitate the specification of repeated pairwise comparisons. For example, a user might compare Southeast colleges (A) to colleges in each of the other regions (B, C, D, etc). She first adds Southeast colleges to the top group shelf. She then clicks on the bottom group shelf to open the sloppy rule menu and enter "Region" to specify the attribute but not the values (Fig. 2g). Multiple groups, each corresponding to the colleges in a region, will be automatically added to the bottom group shelf.

Answering Pairwise Comparison Questions

Here, we provide examples for the seven categories of pairwise comparisons and describe how they are answered.

Repetition — Single (e.g., Compare private to public colleges). As users specify a comparison in the form "compare A to B", Duo classifies the attributes into highly similar and highly different attributes (Fig. 2c). When there are many attributes in a data table, the classification saves effort by removing the need to inspect attributes one by one to determine which of them is similar or different between the two groups. Grounded in Duet [19], attributes are classified using a logistic regression model. We adjusted the model to increase cross validation accuracies. The detailed equations and the improvements are provided as supplemental material.

Besides attribute classification, Duo generates a sentence that describes the specified comparison (Fig. 2b) to help users spot mistakes in the specification. As users hover over an attribute, a visualization is shown along with a short textual description (Fig. 2d-f). The type of visualization shown is determined by the type of comparison (i.e. one to one, one to multiple and multiple to multiple) being conducted.

Repetition — Repeated (e.g., Compare Southeast colleges to colleges in each of the other regions). For repeated comparisons, Duo displays a list of groups similar to the anchoring point (in this case, Southeast colleges) and a list of groups different from the anchoring point. Similar (different) groups are ranked by number of similar (different) attributes. Clicking on a group shows the list of similar or different attributes.

Group — One to One (e.g., Compare MIT to Harvard). A list of similar and different attributes is shown as users specify the comparison. When users hover over an attribute, a visualization along with a textual description is shown (Fig. 2f). For one-to-one comparisons, the visualization displays the values of the selected attribute for the two groups while the text describes how different the two values are.

Group — One to Multiple (e.g., Compare Emory to other colleges in the Southeast). To specify this query, users add `Name=`*Emory University* (base rule) using the top group shelf, and add `Region=`*Southeast* (base rule) and `Name=`*Emory University* (exclusion rule) using the bottom group shelf. For one-to-multiple comparisons, the visualization that pops up upon hovering over an attribute contains a density plot (Fig. 2e) that shows the distribution of multi-object groups (in this case, colleges in the Southeast). By clicking on the density plot, it is converted to a histogram.

Group — Multiple to Multiple (e.g., Compare private to public colleges). For multiple-to-multiple comparisons, the visualization contains two density plots (Fig. 2d), each showing a group's distribution. Users can convert the density plots to a grouped bar graph by clicking on the them.

Attribute — Attribute Absent (e.g., Compare MIT to Stanford). When no attributes are specified, Duo considers all attributes in the data table. In other words, Duo deems *"compare MIT to Stanford"* and *"compare MIT to Stanford on all attributes in the table"* the same. This allows users to compare MIT and Stanford broadly when they have no specific attributes of interest.

Attribute — Attribute Present (e.g., Compare MIT and Harvard on graduate earnings). When users specify some attributes, Duo operates on a truncated data table that contains only the columns corresponding to the specified attributes. Everything else is the same as when no attributes are specified.

5 USER STUDY

Natural language is a clear alternative to specifying pairwise comparison questions due to a short semantic distance [17]: users can directly express their goals in words rather than learning to perform low-level operations as in GUI.

Indeed, much research has been devoted to bringing natural language to data analysis systems such as spreadsheets (e.g., [3]). How natural is natural language when it comes to pairwise comparisons? Does its intuitiveness break at some point? To investigate how sloppy rules stack up against natural language, we conducted a within-subject study during which participants perform pairwise comparison tasks of varying levels of difficulty using the two techniques.

Participants and Apparatus

We recruited 16 graduate students (7 females, ages 22 - 33) through university mailing lists. The inclusion criteria were **1)** native English speakers or **2)** non-native speakers who stayed in the US for at least 2 years. We required participants to be reasonably good at articulating ideas in English so that our results can better generalize to populations with lower English proficiency: if people who are fluent in English perform worse using the natural language interface, people who lack fluency might perform even worse. Of the 16 participants, 11 are native speakers. All participants rated themselves as fully fluent in English. On a 7-point scale ranging from novice (1) to expert (7), the average self-rated level of data analysis experience was 4.875. None of the participants had used any of the interfaces prior to the study. We compensated the participants with a $25 gift card.

The study was conducted in a quiet lab environment using a Macbook Pro with a 13-inch display and a trackpad. During the study, the trackpad rather than a mouse was used for text selection to reduce variability. Non-Mac users were given a short tutorial of basic trackpad operations.

Study Design

Our study was a 2 × 4 within-subjects design in which participants used two interfaces (natural language and sloppy rules) to perform pairwise comparison tasks of four levels of difficulty: easy, moderate, difficult, and very difficult. Participants perform 3 (practice) + 4 (test) tasks using interface A and then 3 (practice) + 4 (test) tasks using interface B. The three practice tasks were the same for both interfaces to give participants an idea that both interfaces could do the same pairwise comparisons. To reduce learning effects, the presentation order of the interfaces was counterbalanced, the tasks were presented in increasing level of difficulty within an interface condition, and two different task sets of four were used across the two interface conditions (but the presentation order of task sets remained the same across participants).

Tasks. We designed a set of three practice tasks and two sets of four test tasks. The attributes and values in the practice tasks and the test tasks come from a car dataset and the college dataset [27] respectively. The two datasets were used as the subject matters were accessible to general audience.

Figure 3: (a) A task of moderate difficulty has 5 rules. The natural language interface (b) rejects nonsense queries, (c) outputs *"YOU GOT IT RIGHT!"* for a correct query, and (d) presents its interpretation for an incorrect query.

The pairwise comparison tasks are illustrated by a visual language (Fig. 3a). The rules for group definition are connected by three operators: **AND**, + and – that correspond respectively to logical conjunction, adding objects to a group and removing objects from a group (the three common operations for group definition). We considered using English sentences to present the pairwise comparison tasks but were aware of its lack of ecological validity: during data analysis, we often form an idea of what to compare before phrasing a pairwise comparison sentence, rather than being told to type a sentence directly to a text box.

Level of difficulty is determined by number of rules: easy, moderate, difficult and, very difficult tasks have 3, 5, 8, and 10 rules respectively. The easy, moderate, and difficult tasks were designed with realism in mind. For instance, the tasks with a moderate level of difficulty (Fig. 3a) only has the **AND** operator, which appears more frequently than + and – in pairwise comparison questions.

Interfaces. A natural language interface was implemented for the study. Following "Explore" in Google Sheets [14], the interface has a data table on the left and provides a text box in the sidebar for entering a question in a single query (Fig. 3b-d). To reduce typing effort, it offers autocomplete suggestions of attribute names and values while users are typing. We originally planned to implement a fully-functioning natural language interface that supports asking any pairwise comparison questions but were concerned about potential inaccuracy in inferring a parse tree from more complex queries.

We concluded that the natural language interface for our study should be able to **1)** recognize a pairwise comparison query if it is correct and **2)** reject a query if it is incorrect.

To achieve **1)**, we crowdsourced pairwise comparison questions by presenting the pairwise comparison tasks in our study to AMT workers. 30 questions were collected for each task, yielding a total of 30×3 (practice) + 30×8 (test) questions. The workers were paid \$0.53 for each question (hourly wage = \$7.25). We then built a logistic regression model using the collected questions. Using the model, the natural language interface classifies participants' queries into one of the pairwise comparison tasks in our study.

For each query, the logistic regression model assigns a probability for each task. To achieve **2)**, the interface rejects queries that are unlikely to be one of the study tasks (Fig. 3b). For a query with a high probability to be one of the study tasks, the interface checks if all rules in the task exist. If there are no missing rules, the interface responds with *"YOU GOT IT RIGHT!"* (Fig. 3c); if some rules are missing, it presents the interpretation of the query so that the participants can correct themselves during the study (Fig. 3d).

The sloppy rule interface was also modified to show *"YOU GOT IT RIGHT!"* for correct queries, and present its interpretation of a query as users specify pairwise comparison rules.

Dependent Variables. There are three dependent measures for our study, the first being the completion time for each task. An end-of-study questionnaire (see the supplemental materials) also collected participants' preferences of interfaces. In particular, two questions were asked for each level of difficulty:

Q1) *Which interface made it easier to specify the pairwise comparison questions (1=interface A much easier, 4=neutral, 7=interface B much easier)?*

Q2) *Which interface would you prefer to perform the pairwise comparison tasks (interface A or interface B)?*

Procedure

Each session lasted approximately 60 minutes. The participants first watched a 5-minute video that introduced the visual language. For the first interface, they watched a tutorial video introducing the interface (~3 minutes) and were instructed to perform the three practice tasks before performing the four test tasks. The participants were told to complete the tasks as fast as possible. During the practices, the experimenter helped the participants overcome difficulties. The participants followed the same sequence for the second interface. For a fair comparison, when the participants used the sloppy rule interface, they were asked to disambiguate an intended rule by typing rather than clicking on an attribute or value from the sloppy rule menu. Their interactions with the interfaces were screen-captured and the completion times were inferred from the videos. After the tests, the participants completed the questionnaire concerning their preferences of interfaces for each level of difficulty. During an end-of-study interview, they were asked about the relative pros and cons of the two interfaces.

Hypotheses

We considered two hypotheses:

H1. Interface interacts with level of difficulty in predicting completion time. In particular, sloppy rules are comparable to natural language for easy tasks and tasks of moderate difficulty, and are more efficient than natural language for difficult and very difficult tasks.

H2. The participants find it easier to specify difficult and very difficult pairwise comparison tasks using the sloppy rule interface but are indifferent between the two interfaces for easy tasks and tasks of moderate difficulty.

Results

Completion Time. The task completion times were log-transformed to approximate normality before the analysis. Using a two-way repeated measures ANOVA, we found a significant interaction between interface and level of difficulty on completion time ($F_{3,45} = 2.95$, $p = .043$). Post-hoc Wilcoxon signed-rank tests with Bonferroni correction indicated that only the time differences for easy ($Z = -2.43$, $p = .015$), difficult ($Z = -2.46$, $p = .014$) and very difficult ($Z = -3.02$, $p = .002$) tasks are significant (Fig. 4b). The results partially align with **H1**. The participants finished the pairwise comparison tasks more efficiently at least for the more difficult tasks.

Ease of Specification. For Q1, we found a significant Spearman's correlation ($r_s = 0.51$, $p < .001$) between level of difficulty and preference (Fig. 4c). The median preference for easy, moderate, difficult and very difficult tasks, were 3.5, 5, 6, and 6.5 respectively (Fig. 4c). Sign tests with Bonferroni correction found that only the median for difficult ($Z = -3.12$, $p = .002$) and very difficult ($Z = -3.30$, $p < .001$) tasks were significant different from 4 (neutral). For Q2, binomial tests indicated that the number of participants who preferred sloppy rules is significantly different from the number of participants who preferred natural language only for the difficult ($p = .004$) and very difficult ($p < .001$) tasks (Fig. 4d). The results support **H2**, indicating a stronger preference for the sloppy rule interface for more difficult tasks.

Pros and Cons of the Two Interfaces. At the end of the sessions, we asked the participants to comment on the relative pros and cons of the two interfaces.

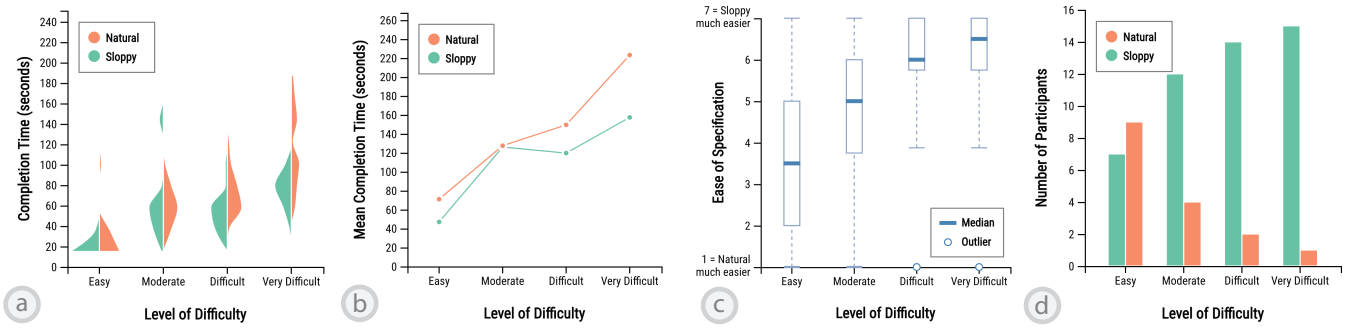

Figure 4: (a) Distributions of completion time at different difficulty levels. (b) Mean completion times for each interface at different difficulty levels. (c) Results of the Likert-scale question (Q1 of the questionnaire). (d) Participants' preference at each difficulty level (Q2 of the questionnaire).

Many participants commented that it was **difficult to articulate long queries** using natural language: *"You have to say things that you wouldn't say in English ... not necessarily you wouldn't say but like it's clunky"* (P10) and *"it became a little bit wordy as they [the pairwise comparison questions] got a little bit longer"* (P7). P8 also expressed the confusion of word choice: *"In natural language, do you use 'and' between two groups [compare A and B] or do you use 'to' [compare A to B] or do you use 'with' [compare A with B]?"* Also, natural language can be **ambiguous** even to humans. P10 commented: *"If I am saying USA cars and Japanese cars with a horsepower greater than 1000 ... Where's the parenthesis?"* His example has two valid interpretations, differing in where to place the parenthesis: 1) cars where (`Origin`=*USA* OR `Origin`=*Japanese*) AND `Horsepower`>*1000*, and 2) cars where `Origin`=*USA* OR (`Origin`=*Japanese* AND `Horsepower`>*1000*).

Despite these challenges, most participants appreciated the **intuitiveness and ease of learning** of the natural language interface: *"If it is a simple task, the second system [the natural language interface] is much more intuitive ... you don't need to learn ... just type it in"* (P5), and *"Interface B [the natural language interface] is nice in a way that it was how you think of the problem"* (P6).

On the other hand, the sloppy rule interface **helps articulate complex pairwise comparison queries**. P6 liked the capability to enter partial rules: *"You just need to type a little bit of what you are thinking and have it complete for you so you don't have to think about how to structure it."* P10 liked how the tabs provide structure for the rules: *"It [the sloppy rule menu] specifies if you are doing 'and' [base rules] or 'plus' [inclusion rules] or 'minus' [exclusion rules] ... you can visually see those in the tabs."*

However, the participants found sloppy rules **harder to learn**: *"I think in interface A [sloppy rules], it wasn't intuitive ... it's not generally how we work in computers ... so it takes some time to get used to that"* (P7), *"There was a bit of a learning curve"* (P8), and *"Navigating between the tabs [using arrow keys] was not intuitive"* (P15). Furthermore, unlike natural language interfaces that can accept different queries (e.g., correlation, and ranking), the sloppy rule interface **only supports pairwise comparisons** and is *"very task-specific"* (P7).

Discussion

The results suggest that the two interfaces are comparable in efficiency and preference for easier pairwise comparison tasks and that the sloppy rule interface is faster and more preferable for more difficult tasks. The qualitative feedback shed lights on two dimensions along which query techniques can be compared: query complexity and learnability. In this section, we discuss implications of query complexity and learnability on designing query techniques for pairwise comparisons and reflect on the limitations of our study.

Query Complexity. The participants' comments reveal two challenges of natural language for more complex queries: articulation difficulty and ambiguity. While the search for more efficient and accurate algorithms to parse natural language reigns supreme in conventional NLP research, these challenges are outside its purview — faster and more accurate natural language parsing do not make complex concepts easier to articulate, neither do they fix the ambiguity in human language. By offering human-centered UI technologies, HCI research is uniquely positioned to tackle these challenges.

Prior research endeavors enhanced our capability to communicate with data via long natural language queries. For example, to showcase DataTone [10], Gao et al. presented a sample query: *"What is the relationship between unemployment and family income for those families earning more than 2000 and less than 150000 between 2007 and 2010 for California and Michigan?"* Conversely, our findings indicate that long complex queries like this should be avoided due to the inherent articulation difficulty. To keep queries short, natural

language interfaces could enable nested conversations [7]. For pairwise comparisons, such interfaces could allow users to say *"create a group of Southeast colleges with admission rate below 0.2"*, *"create a group of New England colleges with admission rate below 0.2"*, and *"compare the first group with the second"*. For spreadsheets, a solution is to provide a button in the toolbar that opens the sloppy rule interface for specifying complex pairwise comparisons.

Although human language is ambiguous, natural language systems could ensure they share a common ground with users by offering an ambiguity widget [10]. For complex pairwise comparisons, the sloppy rule interface could be modified for disambiguation. For instance, after parsing a pairwise comparison question, a system could display the pairwise comparison rules extracted from the question and let users fix the rules using the sloppy syntax if needed.

Learnability. From our study, we have learnt that many participants did not prefer the sloppy rule interface for easier tasks because it is harder to learn. What causes its lower learnability? Lessons from history of user interface design tell us that interfaces that exploit skills and knowledge users already have are more learnable [2]. This is why natural language interfaces are acclaimed: we can express our intent using the language we use in our daily communication, reducing the effort to learn an interface. In contrast, the sloppy rule interface is less effective in leveraging users' prior skills and knowledge. For instance, a participant reported that using the up and down arrow keys to navigate between the **AND**, **ADD**, and **REMOVE** tabs was unintuitive because it did not align with his habit of using shortcuts. He suggested to use the tab key for navigating between the three tabs.

Are natural language interfaces really easy to learn? Existing natural language interfaces are limited by a set of predefined commands. It means that like conventional command lines, users often have difficulty discovering and learning what commands are supported by a natural language interface. This learnability issue was not investigated in our user study because participants were told to ask pairwise comparison questions and hence did not need to discover on their own what types of questions they could ask. To improve learnability, techniques that facilitate learning natural language commands (e.g., a help menu) could be incorporated into natural language interfaces.

Limitations. Aside from not being able to investigate discoverability of natural language commands, our study is limited by using a visual language to present the pairwise comparison tasks (Fig 3a). The visual language, although carefully designed, might favor the sloppy rule interface because of a more direct mapping to the sloppy rule menu. Replication studies would be required to understand how task presentation method moderates the relationship between task performance and query interface.

6 CONCLUSIONS

Many of us share the dream of a world where people can freely communicate with data. Albeit just a tiny step, our work contributes to this vision in three critical ways. We have provided an investigation into the make-up of pairwise comparisons. We have demonstrated how the widely-used spreadsheets can adopt our findings and be endowed with pairwise comparison capabilities. Finally, our comparative study of the sloppy rule interface and the natural language interface provides design implications for data analysis tools that aim to support a wide range of pairwise comparisons.

ACKNOWLEDGMENTS

We thank Richard Catrambone, Jordan Sparks and Bahador Saket for their feedback on the experimental design, Nicoleta Serban for early discussions, and the reviewers for the invaluable comments. Special thanks to the turkers and the user study participants who contributed to this project.

REFERENCES

[1] Michael Bolin, Matthew Webber, Philip Rha, Tom Wilson, and Robert C Miller. 2005. Automation and customization of rendered web pages. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*. ACM, 163–172. https://doi.org/10.1145/1095034.1095062

[2] John M Carroll, Robert L Mack, and Wendy A Kellogg. 1988. Interface metaphors and user interface design. In *Handbook of Human-Computer Interaction*. Elsevier, 67–85. https://doi.org/10.1016/B978-0-444-70536-5.50008-7

[3] Kedar Dhamdhere, Kevin S McCurley, Ralfi Nahmias, Mukund Sundararajan, and Qiqi Yan. 2017. Analyza: Exploring data with conversation. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*. ACM, 493–504. https://doi.org/10.1145/3025171.3025227

[4] Fan Du, Catherine Plaisant, Neil Spring, and Ben Shneiderman. 2016. EventAction: Visual analytics for temporal event sequence recommendation. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 61–70. https://doi.org/10.1109/VAST.2016.7883512

[5] Fan Du, Catherine Plaisant, Neil Spring, and Ben Shneiderman. 2017. Finding similar people to guide life choices: Challenge, design, and evaluation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 5498–5544. https://doi.org/10.1145/3025453.3025777

[6] Michael Yoshitaka Erlewine. 2009. Ubiquity: Designing a multilingual natural language interface. In *Proceedings of SIGIR Workshop on Information Access in a Multilingual World*.

[7] Ethan Fast, Binbin Chen, Julia Mendelsohn, Jonathan Bassen, and Michael S Bernstein. 2018. Iris: A Conversational Agent for Complex Tasks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 473. https://doi.org/10.1145/3173574.3174047

[8] Leon Festinger. 1954. A theory of social comparison processes. *Human Relations* 7, 2 (1954), 117–140. https://doi.org/10.1177/001872675400700202

[9] The R Foundation. 2018. The R Project for Statistical Computing. Retrieved from https://www.r-project.org.

[10] Tong Gao, Mira Dontcheva, Eytan Adar, Zhicheng Liu, and Karrie G Karahalios. 2015. Datatone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*. ACM, 489–500. https://doi.org/10.1145/2807442.2807478

[11] Dedre Gentner and Keith J Holyoak. 1997. Reasoning and learning by analogy: Introduction. *American Psychologist* 52, 1 (1997), 32. https://doi.org/10.1037/0003-066x.52.1.32

[12] Dedre Gentner and Linsey A Smith. 2012. Analogical Reasoning. In *Encyclopedia of Human Behavior*. Academic Press. https://doi.org/10.1016/B978-0-12-375000-6.00022-7

[13] Michael Gleicher. 2018. Considerations for Visualizing Comparison. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 413–423. https://doi.org/10.1109/TVCG.2017.2744199

[14] Google. 2018. Google Sheets - create and edit spreadsheets online, for free. Retrieved from https://www.google.com/sheets/about/.

[15] Matthias J Gruber, Bernard D Gelman, and Charan Ranganath. 2014. States of curiosity modulate hippocampus-dependent learning via the dopaminergic circuit. *Neuron* 84, 2 (2014), 486–496. https://doi.org/10.1016/j.neuron.2014.08.060

[16] Marti A Hearst. 2011. 'Natural' search user interfaces. *Commun. ACM* 54, 11 (2011), 60–67. https://doi.org/10.1145/2018396.2018414

[17] Edwin L. Hutchins, James D. Hollan, and Donald A. Norman. 1985. Direct Manipulation Interfaces. *Human-Computer Interaction* 1, 4 (Dec. 1985), 311–338. https://doi.org/10.1207/s15327051hci0104_2

[18] Johannes Kehrer, Harald Piringer, Wolfgang Berger, and M Eduard Gröller. 2013. A Model for Structure-based Comparison of Many Categories in Small-Multiple Displays. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2287–2296. https://doi.org/10.1109/TVCG.2013.122

[19] Po-Ming Law, Rahul C Basole, and Yanhong Wu. 2019. Duet: Helping Data Analysis Novices Conduct Pairwise Comparisons by Minimal Specification. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 427–437. https://doi.org/10.1109/TVCG.2018.2864526

[20] Greg Little and Robert C Miller. 2006. Translating keyword commands into executable code. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*. ACM, 135–144. https://doi.org/10.1145/1166253.1166275

[21] U.S. News & World Report L.P. 2018. Compare Colleges and Universities. Retrieved from https://www.usnews.com/best-colleges/compare.

[22] Microsoft. 2018. Create a PivotTable to Analyze worksheet data - Office Support. Retrieved from https://support.office.com/en-us/article/create-a-pivottable-to-analyze-worksheet-data....

[23] Robert C Miller, Victoria H Chou, Michael Bernstein, Greg Little, Max Van Kleek, David Karger, et al. 2008. Inky: a sloppy command line for the web with rich visual feedback. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*. ACM, 131–140. https://doi.org/10.1145/1449715.1449737

[24] Christina Niederer, Holger Stitz, Reem Hourieh, Florian Grassinger, Wolfgang Aigner, and Marc Streit. 2018. TACO: Visualizing Changes in Tables over Time. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 677–686. https://doi.org/10.1109/TVCG.2017.2745298

[25] Don Norman. 2007. The next UI breakthrough: command lines. *Interactions* 14, 3 (2007), 44–45. https://doi.org/10.1145/1242421.1242449

[26] Numbeo. 2018. Compare Cities. Retrieved from https://www.numbeo.com/cost-of-living/comparison.jsp.

[27] U.S. Department of Education. 2018. College Scorecard Data. Retrieved from https://collegescorecard.ed.gov/data/.

[28] Sharon Oviatt. 1995. Predicting spoken disfluencies during human-computer interaction. *Computer Speech and Language* 9, 1 (1995), 19–36. https://doi.org/10.1006/csla.1995.0002

[29] Sharon Oviatt. 1996. Multimodal interfaces for dynamic interactive maps. In *Proceedings of the 1996 CHI Conference on Human Factors in Computing Systems*. ACM, 95–102. https://doi.org/10.1145/238386.238438

[30] Sharon Oviatt. 1997. Multimodal interactive maps: Designing for human performance. *Human-Computer Interaction* 12, 1 (1997), 93–129. https://doi.org/10.1207/s15327051hci1201&2_4

[31] Quicksilver. 2018. Quicksilver — Mac OS X at your Fingertips. Retrieved from https://qsapp.com/index.php.

[32] Rajeev Ramchand. 2018. Comparing Suicide Rates: Making an Apples to Apples Comparison. (April 2018). Retrieved from https://www.rand.org/blog/2018/04/comparing-suicide....

[33] Aza Raskin. 2008. The linguistic command line. *Interactions* 15, 1 (2008), 19–22. https://doi.org/10.1145/1330526.1330535

[34] Greg Ridgeway. 2007. *Analysis of racial disparities in the New York Police Department's stop, question, and frisk practices*. Technical Report. Santa Monica, CA, USA.

[35] Richard M Ryan and Edward L Deci. 2000. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology* 25, 1 (2000), 54–67. https://doi.org/10.1006/ceps.1999.1020

[36] Thomas L Saaty. 2008. Decision making with the analytic hierarchy process. *International Journal of Services Sciences* 1, 1 (2008), 83–98. https://doi.org/10.1504/IJSSci.2008.01759

[37] Niloufar Salehi, Lilly C Irani, Michael S Bernstein, Ali Alkhatib, Eva Ogbe, Kristy Milland, et al. 2015. We are dynamo: Overcoming stalling and friction in collective action for crowd workers. In *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems*. ACM, 1621–1630. https://doi.org/10.1145/2702123.2702508

[38] Peggy Sito and Zhang Shidong. 2018. Hong Kong seen winning mega-IPO from Xiaomi after reforming listing rules. (Jan 2018). Retrieved September 1, 2018 from https://www.scmp.com/business/companies/article/2131348/chinese-smartphone-maker-xiaomi-said....

[39] John W Tukey. 1949. Comparing individual means in the analysis of variance. *Biometrics* (1949), 99–114. https://doi.org/10.2307/3001913

[40] Paul Vegas. 2018. Barcelona defender Mina to decide between Man Utd, Everton today. (Aug 2018). Retrieved September 1, 2018 from http://www.tribalfootball.com/articles/barcelona-defender-mina-to-decide-between-man-utd....