Source Selection Languages: A Usability Evaluation

Ixent Galpin
Dpto. de Ingeniería
Universidad Jorge Tadeo Lozano
Bogotá, Colombia
ixent@utadeo.edu.co

Edward Abel
School of Computer Science,
University of Manchester
Manchester, UK
edward.abel@manchester.ac.uk

Norman W. Paton
School of Computer Science,
University of Manchester
Manchester, UK
norman.paton@manchester.ac.uk

ABSTRACT

When looking to obtain insights from data, and given numerous possible data sources, there are certain quality criteria that retrieved data from selected sources should exhibit so as to be most fit-for-purpose. An effective source selection algorithm can only provide good results in practice if the requirements of the user have been suitably captured, and therefore, an important consideration is how users can effectively express their requirements.

In this paper, we carry out an experiment to compare user performance in two different languages for expressing user requirements in terms of data quality characteristics, *pairwise comparison of criteria values*, and *single objective constrained optimization*. We employ crowdsourcing to evaluate, for a set of tasks, user ability to choose effective formulations in each language. The results of this initial study show that users were able to determine more effective formulations for the tasks using pairwise comparisons. Furthermore, it was found that users tend to express a preference for one language over the other, although it was not necessarily the language that they performed best in.

CCS CONCEPTS

• **Information systems → Mediators and data integration**; • **Applied computing → Decision analysis**;

KEYWORDS

Information integration, data wrangling, source selection, data quality, decision analysis

ACM Reference Format:
Ixent Galpin, Edward Abel, and Norman W. Paton. 2018. Source Selection Languages: A Usability Evaluation. In *HILDA'18: Workshop on Human-In-the-Loop Data Analytics, June 10, 2018, Houston, TX, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3209900.3209906

1 INTRODUCTION

Data scientists are tasked with obtaining insights from data. However, in many domains there are numerous data sources for the same kinds of data, and likely the cost of wrangling all such sources would be prohibitive. Hence, the problem of *source selection*, where a subset of the available sources is identified, has become an area of active investigation (e.g. [1, 4, 8–10, 12]).

Source selection methods have to choose between sources based on certain criteria. In some cases specific criteria are encoded within the source selection technique. For example, the *marginal gain* has been proposed as a criterion for determining when the value of an additional source exceeds the cost of incorporating it, where the value of data is based on quality metrics [4]. However, the use to which the data is to be put likely influences the fitness-for-purpose of a data source, and other researchers have the human-in-the-loop for source selection, for example applying multi-criteria decision support methods that allow the user to indicate which criteria are more or less relevant to them [1, 9, 12].

Such techniques are then evaluated empirically, investigating various features of the source selection method. For example, experiments have evaluated techniques based on their profitability [4], result utility [1], execution time [1, 4] and location within a multi-objective trade-off space [1, 12]. Such evaluations provide insights on the effectiveness of the method given some criteria, but do not provide insights into the usability of the approach.

As a result, an open question is *how effectively can users express their source selection requirements using different languages*. An effective source selection algorithm can only provide good results in practice if the requirements of the user have been suitably captured. In this paper we report on an experimental study that compares two languages for expressing source selection requirements: *pairwise comparison of criteria values* (PCCV), and *single-objective constrained optimization* (SOCO). We chose these languages because they offer two commonly used and contrasting approaches for eliciting and expressing user preferences: PCCV have been utilized within a plethora of applications [15] as has SOCO [14]. Moreover, both languages have been used in diverse source selection proposals, e.g. PCCV is used in [1, 12], and SOCO in [4, 8, 10].

In this paper, users recruited through crowdsourcing have been asked to choose between alternative ways of expressing source selection requirements, for a diverse collection of data selection tasks. The quality of the results, regarding users' abilities to choose the most effective formulations for each task in the different languages was analyzed, as well as and in relation to, subjective user preference of each language.

This initial study shows that users were much more able to determine effective formulations for the tasks using pairwise comparisons, whereas they tended to prefer one language over the other, with some users declaring the most usable to be at odds with the language they performed best with. These results suggest that further work on the usability of source selection methods will be important to the construction of effective source selection systems.

HILDA'18, June 10, 2018, Houston, TX, USA

ACM ISBN 978-1-4503-5827-9/18/06…$15.00
https://doi.org/10.1145/3209900.3209906

The paper is structured as follows: Section 2 overviews technical background; Section 3 details the experiment design; Section 4 presents the experimental results and evaluations; and Section 5 concludes.

2 BACKGROUND

Before detailing our experiment design we present technical background relating to the two languages under evaluation.

2.1 Pairwise Comparison of Criteria Values

Pairwise comparison of criteria values (PCCV) is employed in several Multiple-Criteria Decision Analysis (MCDA) methodologies, such as Analytic Hierarchy Process [11], for systematic and transparent elicitation of preferences over a set of criteria. PCCV enables consideration of a pair of criteria at a time, to allow a user to define their preference, and strength of preference, between the pair. Allowing a user to consider only a pair of elements at a time induces a separation of concerns that helps to achieve an accurate reflection of user preferences [13].

Given a pair of criteria, the user is asked which is most important, and by how much, through a verbal scale of descriptive importance (`equal importance to`, `very strong importance over`, etc.). Using a verbal scale is argued to be intuitively appealing and user friendly [6], and precise understanding of the quality criteria's underlining computations is not required. From a set of comparisons a set of numerical criteria weights can be derived through, for example, the use of the geometric mean method widely used in the MCDA community [7]. Such weights can then be utilized within an optimization framework, for example, a simple weighted sum model, or via more intricate optimization techniques [1], to select data that is most aligned with the user's preferences.

For example, given a set of three criteria from the real-estate domain that we will consider in the paper, viz., `priceQuality`, `locationQuality`, `roomInformationQuality`, an example formulation might be:

priceQuality is of very strong importance over **locationQuality**
priceQuality is of equal importance to **roomInformationQuality**
roomInformationQuality is of very strong importance over **locationQuality**

In this case, the user's preferences are that both price and room information quality are (i) both highly more important than location information, and (ii) of equal importance in relation to each other, to the user.

2.2 Single Objective Constrained Optimization

Single objective constrained optimization (SOCO) is inspired by classical linear programming [3]. A single criterion is specified as an optimization goal, in the context of zero or more constraints cast as inequalities in terms of the available criteria. Any records that do not meet any of the constraints are excluded from the final result. Records which have the most favorable value with respect to the criterion specified in the optimization goal are preferred in the final result. As an example, the formulation

```
maximize roomInformationQuality
subject to priceQuality = 1
and locationQuality >= 0.5
```

Street and town exists	1
Only town exists	0.5
Only street exists	0.2
Neither street or town exist	0

Table 1: Calculating the `locationQuality` measure.

states that the records selected in the result must have a `priceQuality` equal to 1, and `locationQuality` greater than or equal to 0.5, and, moreover, that records with the highest `roomInformationQuality` should be sought. For a user to employ SOCO in a meaningful manner, a precise understanding of the quality criteria thresholds for the constraints is required. An example of how we compute `locationQuality` is presented in Table 1; the other criteria are computed similarly.

3 EXPERIMENT DESIGN

To gain an insight into the effectiveness and usability of the proposed languages for expressing selection requirements, we carried out an experiment to compare user performance when selecting statements in each of the languages, and also gathered user opinions about the usability of each language.

Overview We prepared a questionnaire that comprises different kinds of questions, viz., *validation questions*, aimed at ascertaining respondent (henceforth referred to as "user") reliability, *language performance questions*, to collect evidence about user effectiveness when employing a language, and *usability questions*, used to obtain user opinion about a language.

The questionnaire initially presents an introduction to the dataset used, and provides an explanation of the criteria of interest, coupled with a validation question to check the user's understanding. Subsequently, a short tutorial is presented for one of the languages, and four tasks are given to the user. After completing the tasks, the user is asked four multiple-choice questions about his/her opinion of the language, adapted from the well-established System Usability Scale (SUS) questionnaire [2]. The same process is repeated for the other language (both language orders were experimented with, to prevent any variation in the results that language order may cause). Finally, some qualitative data is collected, in which the user is asked about the language he/she preferred and the reasons.

Tasks Each task comprises two parts. Firstly, given a natural language description of the data that is required, the user is required to choose, among four possible formulations, the one deemed best to obtain the data with the quality characteristics required. Note that the formulations presented to the user are selected such that they yield results of starkly contrasting quality. Secondly, in order to verify sufficient understanding of the task description (and also to validate against random selection of answers), the user is presented with the result sets associated with the formulations shown previously, and asked which one would best satisfy the requirements.

The types of tasks selected for the experiment are intended to be varied and have diverse characteristics, in order to cover a broad range of possible user requirements. Table 2 presents a description of the tasks used in generic, domain-independent terms, alongside an example task using the real estate domain. Given that the languages have different functionalities, having diverse tasks

enables usability to be evaluated in interesting cases such as when one language is, in principle, more suitable for a task than the other, e.g. PCCV are potentially less well-suited for expressing *Task 4*.

Evaluation Measures The validation and language effectiveness questions employ the notion of an ***Answer Rank*** for each question. For each formulation, the utility of its associated result set can be determined. The utility is based on the F-measure, for which the notion of a true positive depends on the the natural language task description. For example, when seeking data to carry out a price comparison, a true positive would be a record for which complete price information is present. In this way, the F-measure of a result set determines how fit-for-purpose it is. We use F-measure because it provides a succinct means to compare result sets that may be of different sizes. Each set of four language formulations for a task results in distinct utility scores, from which an ordinal ranking of the four can be determined. This results in an answer rank of between between 1 (best answer) and 4 (worst answer)[1].

User reliability (%) This measure aims to determine the degree of understanding and engagement of the user. It is computed using only validation questions, assigning 3 points for each *Rank 1* answer selected, and 1 point for each *Rank 2* answer selected. No points are awarded if a user selects a *Rank 3* or *Rank 4* answer.

User Performance (%) This measure calculates the overall effectiveness of a user at choosing the best formulation for a given language overall all the tasks. It is calculated using answer rank in a similar way to user reliability, using only language performance questions. As such, a user performance score of 100% (for a language) represents a user who selected *Rank 1* formulations for each task.

Usability Score This measure aims to capture subjective user preference about each language. It results in a score between 0 and 100, calculated in a similar manner to SUS.

4 EVALUATION RESULTS

4.1 Experiment Setup

We curated a real-world dataset consisting of web-scraped data from the real-estate property domain, extracted via the DIADEM system [5], and created a questionnaire using Google Forms with the four concrete tasks presented in Table 2. Each task is given once for each language, resulting in eight tasks in total. Two variations of the questionnaire were created, in which the languages are presented in a different order. Respondents to the questionnaires were recruited using the Amazon Mechanical Turk[2] crowd-sourcing platform. A Human Intelligence Task (HIT) was set up within Amazon Turk as an external link to the questionnaire. In order to attract high quality responses, the HIT was made available to people with the Amazon Turk Masters qualification, and those who self identified as IT workers. Participants were paid the UK minimum wage pro-rata according to the anticipated HIT duration time[3]. A custom qualification was created in Amazon Turk and awarded to each respondent who completed the HIT, in order to ensure nobody carried out the HIT twice. Furthermore, we established a threshold for task comprehension. Participants with a user reliability of less than 70% were excluded from the analysis, i.e., we only included results for users who, at worst, gave two *Rank 3* or *Rank 4* answers, and two *Rank 2* answers, for the nine validation questions. In this way, we ensured that we only analyzed results from participants who demonstrated sufficient understanding of the tasks. Of 34 respondents overall, 20 passed the reliability threshold. From these, we discarded two participants at random to ensure equal numbers of participants for both language orders. This resulted in a set of 18 participants overall for analysis.

[1]Depending on the suitability of a language for a given task, it may not be possible for a *Rank 1* answer to have a result set with an F-measure of 1. However, we are interested in evaluating participants' abilities at choosing a language's most suitable formulation.
[2]https://www.mturk.com/
[3]The UK minimum wage for 2018 is 7.83 GBP/hour.

4.2 Results

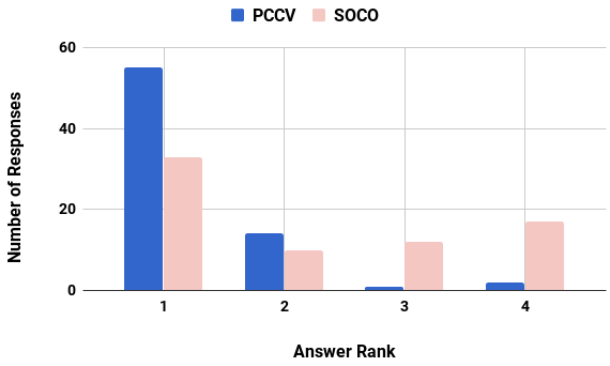


Figure 1: Answer ranks for each language (all tasks).

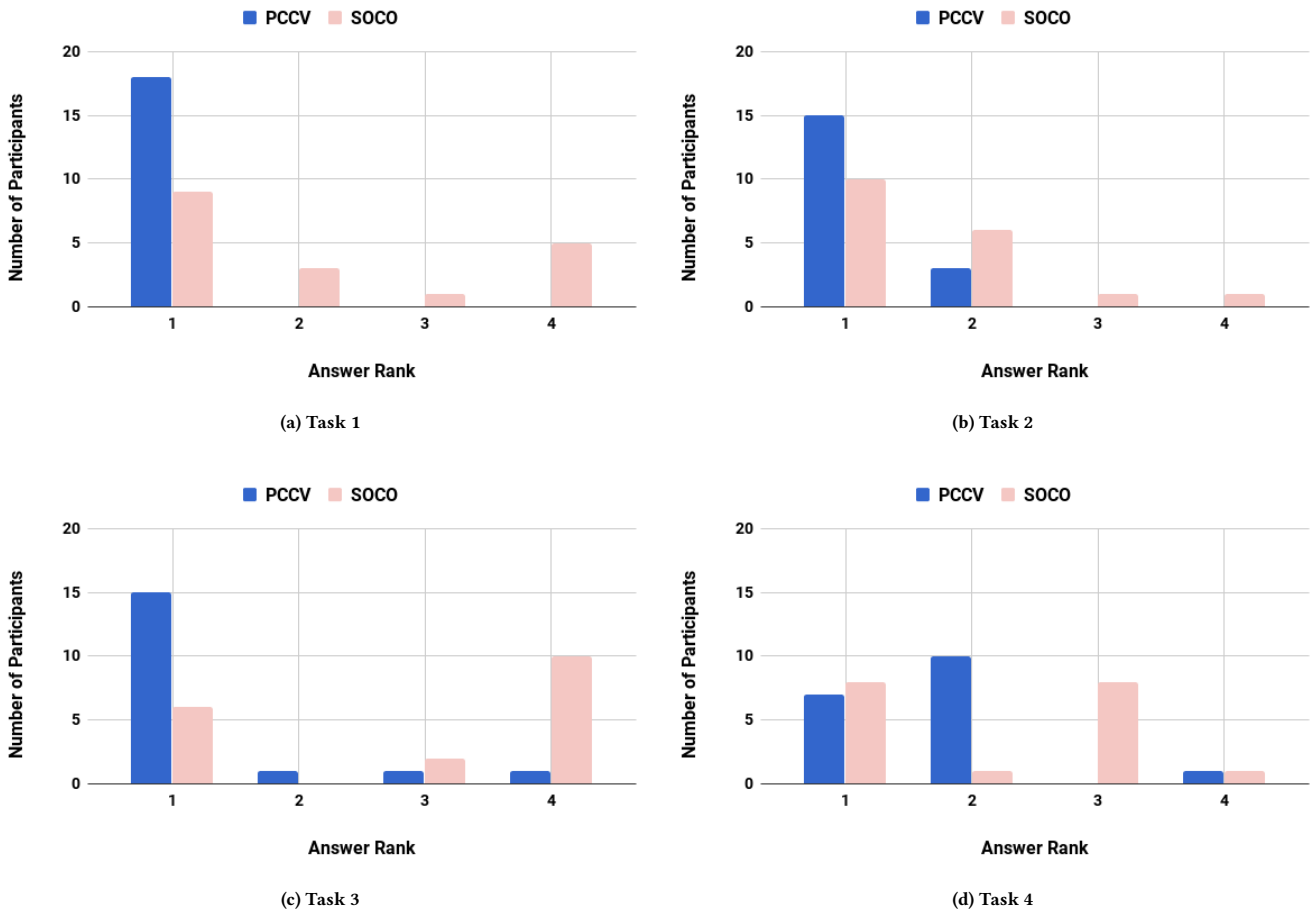
User Performance (All Tasks) Figure 1 presents the number of responses, grouped by answer rank, for each language, combined for all tasks. From this plot we observe a clear difference between the languages regarding our participants' ability to choose the best formulation for the tasks. For PCCV, users chose the top-ranked formulation over 75% of the time, and chose one of the two top-ranked formulations over 95% of the time. Conversely, for SOCO, although users chose the top ranked formulation more than the others, it was only in 45% of cases. The third or fourth ranked formulation was chosen over a third of the time.

With regards to user performance, for PCCV, all users scored between 33% and 100%, and the average was 83%, whereas for SOCO users scored between 8% and 100%, and the average was only 50%. We employed a paired, two tailed, t-test to determine whether there is a significant difference between the user performance for both languages: this gave a p-value of 0.000498, indicating that difference in user performance across both languages is extremely statistically significant, being well under the commonly used 0.05 threshold.

User Performance (By Task) Figure 2 shows the answer rank results broken down by task. For *Tasks 1–3* we observe that most users chose the top-ranked formulation for PCCV (and indeed, for *Task 1*, in all cases chose the top-ranked formulation). However, for SOCO the best formulation was not chosen so often, and selection of all four formulations occur. For *Tasks 1* and *2* the performance p-values obtained are 0.000927 and 0.0757 respectively, indicating that PCCV language performance is significantly greater for *Task*

Task	Abstract Task	Concrete Task
1	*Obtain data that has a certain quality threshold for a single criterion.*	You have a property dataset with poor location data. However, you need to obtain data with the best location quality possible. Ideally, you should obtain as many records as possible that have some location information (even if it is just the street name).
2	*Obtain data that has a certain quality threshold for two given criteria.*	Due to falling sales of properties, you need to compare price information between properties with a similar location. Preferably, you should aim to get both town and street location data. If that's not possible, just the town will do.
3	*Obtain data that is of high quality for at least one of two conflicting criteria.*	You have a property dataset in which most records have high quality data for only one quality measure. For example, if a record has high quality price data, it is unlikely to have high quality location or room information. You wish to obtain some records that have high quality data for either location or room information. You are not worried about price data.
4	*Obtain data that is of high quality for one criterion, but of low quality for another criterion.*	You have a property dataset, and wish to retrieve records with good price information. However, you do not wish to retrieve any records with complete location data.

Table 2: Generic task descriptions with the corresponding domain-specific task.



(a) Task 1

(b) Task 2

(c) Task 3

(d) Task 4

Figure 2: Answer ranks for each language, broken down by task.

1, but not in *Task 2*, which has two criteria to consider. For *Task 3*, the SOCO *Rank 4* formulation was chosen most often, resulting in the task with the worst user performance for this language (33%). It is the task with the most statistically significant difference in user

performance for both languages (with a p-value of 0.000401). It is likely that this was caused by SOCO not supporting disjunction explicitly. For *Task 4*, we observe that users had greater difficulty in choosing the best formulation for PCCV than for *Tasks 1–3*. It is the only task in which the number of *Rank 1* answers given for SOCO is greater than that PCCV. However, we still observe that selection of all four formulations occurs for SOCO. Although user language performance for PCCV remains higher than for SOCO, the p-value obtained of 0.381 reveals that for this result we did not obtain a significant difference in performance. Indeed, this task is not well-suited for PCCV, which does not include a mechanism for certain exclusion of data with certain characteristics. Overall, these results show that PCCV language performance is significantly greater for tasks that are expressible in PCCV (as one would expect), and less significantly so in the case of *Task 4*, which can be suitably expressed in SOCO.

Cross-Language User Performance Comparison In Figure 3(a), we compare user performance for each of the languages. In this plot, point color denotes the language preferred by the user according to the usability score[4]. In this plot, we observe that for most users, high performance in one language results in high performance in the other. Overall the Pearson correlation coefficient between the performance of the two languages is 0.336. However, there is also a cluster of users in the bottom right who performed well with PCCV, but not so well with SOCO. One user scored 100% with PCCV, yet 0% with SOCO, and several cases can be observed of users scoring greater than 66% performance with PCCV, and 33% or less with SOCO. Despite this, and somewhat surprisingly, a couple of these users did not give PCCV a higher usability score, perhaps indicating that they were unaware of their inferior performance with SOCO. Conversely, we do not observe any cases of the reverse, i.e. of users performing well with SOCO, but not so well with PCCV, providing further evidence that PCCV is easier to learn and understand than SOCO.

Cross-Language Usability Score Comparison The usability scores of the languages by user are compared in Figure 3(b), where each point denotes a user's usability scores for the languages, and its series denotes whether the user performed better with PCCV or SOCO[5]. From this we observe a spread of usability scores between the languages from the users, and the plot suggests little correlation between usability scores and the language users performed better with.

Overall, the average usability score was 61 for PCCV, and 53 for SOCO, with a p-value of 0.479. This suggests a very similar overall preference between the languages from the users with only a slight preference for PCCV. However, from this plot we observe that only 17% of users had better performance with SOCO, compared with 83% for PCCV. Interestingly, we observe a cluster of users on the left of the plot, denoting higher usability scores for SOCO than PCCV, despite the majority of them performing better with PCCV. This confirms what we observed in Figure 3(a), i.e. that no users performed significantly better with SOCO than PCCV.

Language performance vs. Usability In Figure 4 we compare user performance with a language to his/her usability evaluation score for the language, to compare each user's performance and perception of each language. Figure 4(b) presents the SOCO performance and usability scores for each user. From this plot, we observe that the better a user performed with the language, the higher his/her usability score for it and the Pearson correlation coefficient here is 0.446, suggesting that user evaluations of SOCO usability is aligned with his/her performance. The results comparing the performance and usability evaluation scores for PCCV for each user in shown in Figure 4(a). Here, we observe less variation in performance for PCCV, compared to SOCO, and that there is little correlation between a user's performance and evaluation of the language

Qualitative analysis Finally, we can glean further comparisons between the languages from qualitative analysis of the user thoughts regarding language usability. Some of the user responses indicating the language they preferred, and the reasons, were:

- *"[PCCV] since it was natural speech with no numeric information"*
- *"[PCCV as I] simply prefer words over numbers"*
- *"[SOCO] because its specific numbers made it easier to understand".*
- *"[SOCO] results could be narrowed down with precision."*

These answers capture how, as we observed from the usability analysis, some users found PCCV more usable whilst others SOCO, and helps to explain why.

5 CONCLUSIONS

Many organizations report that they are struggling to make the most of the available data[6]. Furthermore, the significant cost of data preparation activities means that judicious selection of data sets for wrangling is important to the cost-effectiveness of data analyses[7]. This in turn raises the question *how can data scientists and engineers best express their requirements when selecting the most suitable data sources.* This paper reports on a preliminary user study that shows significant differences in performance with two candidate source selection languages.

The main findings were that: (1) overall user performance was higher for PCCV than SOCO, suggesting that this language is easier to learn and understand; (2) performance in one language tends to be correlated with performance in the other language; (3) there is little correlation between usability scores of each language, indicating that users tend to prefer one language or the other; and (4) for SOCO, user performance tends to be correlated to usability score; this is not the case with PCCV, indicating that users who perform well in this language may in fact prefer SOCO.

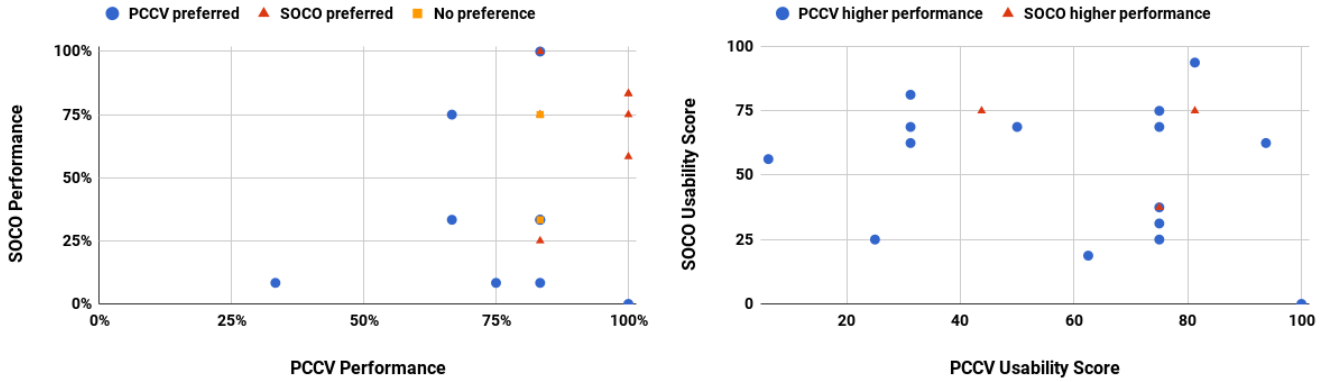
To further enhance understanding of techniques for source selection, further studies could usefully investigate: which criteria are most important for specific applications; whether features of both languages could be combined to create a hybrid language that enables users to express requirements more effectively; what user interfaces are most suitable for capturing user preferences; and how the system can communicate to the user the outcomes that

[4]Users with the same performance for both languages and who are in the same series are shown as a single point.

[5]Users who awarded the same usability score to both languages and are in same series are shown as single point.

[6]https://news.sap.com/sap-study-reveals-key-data-challenges-and-opportunities-in-enterprise-data-landscapes/

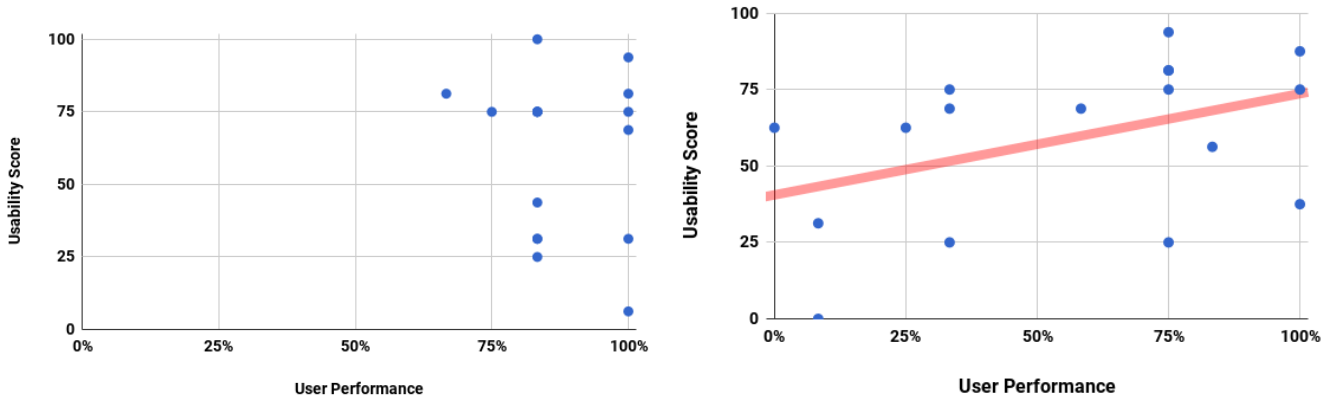
[7]https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#33d344256f63



(a) User performance

(b) Usability score

Figure 3: Comparing performance and usability across both languages.



(a) PCCV

(b) SOCO

Figure 4: User performance vs. usability score

are available and the trade-offs between criteria.

Acknowledgments. This work is supported by the VADA Programme Grant of the UK EPSRC, grant number EP/M025268/1.

REFERENCES

[1] Edward Abel, John Keane, Norman W. Paton, Alvaro A.A. Fernandes, Martin Koehler, Nikolaos Konstantinou, Julio Cesar Cortes Rios, Nurzety A. Azuan, and Suzanne M. Embury. 2018. User driven multi-criteria source selection. *Information Sciences* 430-431 (mar 2018), 179–199.

[2] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.

[3] George B Dantzig. 1951. Maximization of a linear function of variables subject to linear inequalities. *New York* (1951).

[4] Xin Luna Dong, Barna Saha, and Divesh Srivastava. 2012. Less is more. *Proceedings of the VLDB Endowment* 6, 2 (dec 2012), 37–48.

[5] Tim Furche, Georg Gottlob, Giovanni Grasso, Xiaonan Guo, Giorgio Orsi, Christian Schallhart, and Cheng Wang. 2014. DIADEM: Thousands of Websites to a Single Database. *PVLDB* 7, 14 (2014).

[6] Alessio Ishizaka, Dieter Balkenborg, and Todd Kaplan. 2010. Influence of aggregation and measurement scale on ranking a compromise alternative in AHP. *J Oper Res Soc* (2010), 1–11.

[7] Alessio Ishizaka and Phillipe Nemery. 2013. *Multicriteria decision analysis: methods and software.* Wiley.

[8] George A Mihaila, Louiqa Raschid, and Maria-Esther Vidal. 2000. Using Quality of Data Metadata for Source Selection and Ranking.. In *WebDB*. 93–98.

[9] Theodoros Rekatsinas, Amol Deshpande, Xin Luna Dong, Lise Getoor, and Divesh Srivastava. 2016. SourceSight: Enabling Effective Source Selection. In *ACM SIGMOD*. 2157–2160.

[10] Julio César Cortés Ríos, Norman W Paton, Alvaro AA Fernandes, and Khalid Belhajjame. 2016. Efficient feedback collection for pay-as-you-go source selection. In *SSDBM*. ACM, 1.

[11] T. L. Saaty. 1980. *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation.* Mcgraw-Hill. 287 pages.

[12] Jing Tian, Dan Yu, Bing Yu, and Shilong Ma. 2013. A fuzzy TOPSIS model via chi-square test for information source selection. *Knowledge-Based Systems* 37 (2013), 515–527.

[13] Janine van Til, Catharina Groothuis-Oudshoorn, Marijke Lieferink, James Dolan, and Mireille Goetghebeur. 2014. Does technique matter; a pilot study exploring weighting techniques for a multi-criteria decision support framework. *Cost effectiveness and resource allocation : C/E* 12, 1 (jan 2014), 22.

[14] Robert J. Vanderbei. 2001. Linear Programming: Foundations and Extensions. (2001).

[15] Luis G. Vargas. 1990. An overview of the analytic hierarchy process and its applications. *European Journal of Operational Research* 48, 1 (1990), 2–8.