# Evaluating the Suggestiveness of Command Names

**Jarrett Rosenberg**
Department of Psychology
University of California, Berkeley, CA 94720
and
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304

An important feature of the design of human–computer interfaces is that of command languages: the vocabulary and syntax that allow a user to express commands to the system. If we look at command languages from the standpoint of natural languages, rather than formal ones, then there are three aspects to their user interface. The first is the overall structure of the user–system dialogue—its *pragmatics*, so to speak (e.g., [3]), which includes issues of contextual reference, presuppositions, and so on. The second aspect of command languages is their *syntax* (e.g., [1], [4]). The important issue here is the trade–off between consistency of the syntax and its similarity to that of natural language. The third aspect of command languages is their *semantics*, primarily that of their commands. Most command languages are fairly small, with simple data and control structures, and so their semantics are fairly trivial. More important is the "lexical" semantics of commands and their arguments and parameters. The crucial factor here is the names given to the entities and operations in the system by the command language: if those names are not apt, performance will be impaired just as with poorly designed syntax or dialogue structure. This paper investigates the psycholinguistic aspects of this naming problem.

## THE GOODNESS OF COMMAND NAMES.

***Defining goodness.*** Opinions can vary considerably about which names are good or bad, and what makes them so. Compounding the problem is the lack of any objective definition or measure of the "goodness" of a command name. There are a number of different criteria by which the goodness of a command name may be judged: they reflect different aspects of the interaction between the user and the system. The two principal aspects are (1) the connection users make from names to the actions they denote, and (2) the (reverse) connection from an action to be performed to the system's name for it. If the latter aspect is the object of study, then goodness will be defined, for example, in terms of how easy it is for subjects to recall the name of a command when they want to use it. This lends itself to the sort of paired–associate learning and memory analysis ably done by Black and Moran [2]. However, the definition of goodness I am using is based on the former aspect, namely, I claim that *in naming commands we want to maximize the ability to convey an implicit model (i.e., set of relationships) of the system's actions by naming its commands to reflect that model.* For example, if two commands are operational inverses, their names should be linguistic inverses. Note that this says nothing directly about learnability or memorability of the commands; theoretically it should be possible for a suggestive set of names to be poorly remembered, or the reverse, although Black and Moran's work indicates that "discriminability" may be a common element in both notions of goodness. In what follows, it should be kept in mind that I am using "goodness" in the sense of "suggestiveness."

From this name-to-action approach, it follows that a command name is good (with respect to a set of names and a system) to the degree that

1. it directly suggests what the command does (e.g., **print** is better than **list**),
2. it directly suggests the relationships (e.g., similar, opposite, unrelated) of that command to the others in the system.

This "suggestion" process is through a correspondence between the semantics of the name and the semantics of the command. While the latter can be rather precisely specified, the former may not be. However, since compositional semantics in this domain are minimal, it is possible to approximate meanings by the use of simple semantic features. The result is an approximate but efficient representation of the meanings of command names.

The next step is to explicate the nature of the suggestive correspondence and the meaning of "direct". I will claim here that an appropriate interpretation of "direct suggestion" is that of *similarity*. That is, a name is good if its meaning (set of features) is similar to the meaning of the command, and the set of names is similar in semantic structure to the set of commands.

***Goodness as similarity.*** Tversky [5], [6] has proposed an axiomatic set-theoretic model of how similarity judgments are made on the basis of comparison of sets of features. In his model, objects are represented as collections of features. The features are not limited to binary or nominal variables; they are also applicable to ordinal or cardinal variables, such as scales. The model (actually a class of models) may be summarized as follows:

Let $\Delta = \{a, b, c,...\}$ be the domain of objects under study. Let A, B, C denote the sets of features of *a*, *b*, *c*, respectively. Let $\mathbf{s}(a,b)$ be a measure of the similarity of *a* to *b*. Then according to Tversky, the similarity of two objects *a* and *b* is a function of three arguments: $A \cap B$ (the set of features belonging to both *a* and *b*), A–B (the set of features belonging to *a* but not to *b*), and B–A (the set of features belonging to *b* but not to *a*):

$$\mathbf{s}(a,b) = \theta \mathbf{f}(A \cap B) - \alpha \mathbf{f}(A-B) - \beta \mathbf{f}(B-A) \qquad (1)$$

composed of

(i) an (additive) function **f**, which is a measure (counting) of the common ($A \cap B$) or distinctive (A–B, B–A) features of *a* and *b*, and

(ii) $\theta$, $\alpha$, and $\beta$, which are weightings of the importance of the three arguments in a given judgment.

This gives us a basis for an algorithm for computing the similarity among two feature representations, and thus we have a precise explication for the first part of the definition of goodness. (For completeness, one can elaborate Tversky's model by considering the possible role of those features which are not possessed by either *a* or *b*, i.e., $A' \cap B'$. In the research reported here, such an elaboration was not necessary.)

The second part of the definition of goodness is a bit more complicated, since the notions of structure and similarity between structures are very broad. A simple approximation can be had by defining the structures of the names and commands as matrices of within-domain similarity values, and defining the similarity between such structures as the correlation between the matrices.

There is one last step needed here. We want a measure of similarity that contains an implicit and standardized comparison, and one which will enable us to compare similarities across different sets of name–command pairs; the solution is thus to create a sort of *"standardized" similarity*. This can easily be done by taking the ratio between the actual similarity observed between two items and their maximum possible similarity; this reflects how similar the two things are, given how similar they could possibly be. The simplest way to do this to imagine that those features which make the items different were actually the same, i.e., that every feature in A–B and B–A would instead belong to $A \cap B$. Thus given the computed actual similarity of two items, the maximum possible similarity between them is obtained simply by adding rather than subtracting $\alpha \mathbf{f}(A-B)$ and $\beta \mathbf{f}(B-A)$ to the value of $\theta \mathbf{f}(A \cap B)$. The standardized similarity of two objects is then the ratio of the actual and the maximum possible similarities, and is scaled to range between 0 and 1.

## VALIDATING THIS APPROACH.

To test the validity of this approach, two experiments were conducted. The first involved having 12 computer-naive subjects make judgments about the semantic similarity of sets of command names (considered as ordinary verbs). Analogous judgments about the functional similarity of a set of 10 text-editor commands were then collected from four programmers. From a comparison of the two sets of judgments the goodness of the names for those commands was computed.

In the second experiment, another group of 12 computer-naive subjects were presented with the command names and a set of before-after pictures representing the actions of the text-editor commands. The goodness of each name was used to predict how accurately subjects could pick the correct command-action when given its name. This task was chosen as a validation task for two reasons: first, it most directly tests the notion of goodness as suggestiveness, and second, it is similar to the actual situation in some computer systems where a user may see a list of all commands available in a given context. Accurate selection from such a menu depends on the goodness of the names as defined here. Consider the poor novice searching such a list for the name of the TENEX command for printing a document: it's called *list*.

***Measuring goodness — Experiment 1.*** Ten commonly used text-editor commands (from the UNIX line-oriented editor *Ex*) and three different sets of names for them were used in the research. The triads method was used for eliciting semantic judgments: subjects were given triples of words and asked to pick which two were more related, and to give a brief justification. This produced two kinds of information: first, a similarity matrix could be constructed from the pairings, with the similarity of two words being a function of how frequently they were paired with each other (opportunities for each pairing occurred equally often). Second, a set of features and their assignment to the words could be extracted from the justifications.

Each of the 12 subjects (computer-naive college students) was given two of the three namesets to judge (separated by a brief problem-solving task for a break). For each nameset 48 of the possible 120 triples were used; each pair of words occurred at least three and at most four times. The task was not difficult for subjects to do, although they found it fatiguing. The four programmers performed the triads task using the command actions, judging their functional similarity.

A total of 40 features were mentioned by the computer-naive subjects, although only 13 were mentioned often enough to be considered useful for calculating featural similarity. Agreement among computer-naive subjects, as measured by the average correlation among their pairing judgments, was low ($r$ = .47, .48, and .44 for the three namesets). In contrast, the programmers' pairings of the commands showed high agreement ($r$ = .85), and only six features were used with high frequency.

Using the features collected, goodness values for each name were computed using Tversky's linear contrast model. Pilot research of mine and that of Tversky (personal communication) suggested that shared features were more important than distinctive ones, and that there was a symmetry in the effect of distinctive features, so the coefficients $\theta$, $\alpha$, and $\beta$ in formula (1) were set to 2, 1, and 1, respectively. Call this the symmetric model of goodness. An alternative weighting is suggested by the idea that the mapping between the meanings of name and command is asymmetric: for a name to be good, what is important is that its meaning subsume that of the command; thus if the two sets of features are not equivalent, it is more important that the features of the command be a subset of the features of the name than the reverse (although if there are many additional features in the name, it can become misleading). Thus distinctive features of the command detract more from the goodness of a name, and the weighting suggested is one of 4, 1, and 2. Call this the asymmetric model.

***Predicting accuracy from goodness — Experiment 2.*** In the second experiment, each of 12 computer-naive subjects was given all three namesets to work with. On each page of the test booklet there were 12 pairs of before-after pictures depicting the actions of text-editor commands. The first 10 corresponded to the 10 editor commands, while the last two were "distractor" pictures for the commands **print** and **write** in Nameset 1. For each nameset, subjects first were presented with a different name (but the same set of pictures) on each page, and told to pick which picture or pictures, if any, it went with. After judging all 10 names separately, they were presented with the list of all 10 names, and asked to pair the names with the pictures in a one-to-one fashion, i.e., each name was to go with one and only one picture. It was hypothesized that the latter condition would increase accuracy, since the one-to-one constraint would allow certain possibilities to be ruled out, thus counteracting to some degree the badness of a name.

There were three measures of accuracy: correctness in first choice of picture, correctness in all choices (i.e, whether the set of all the subject's choices for a name contained the correct picture), and correctness in the constrained condition where the whole list was judged together. These will be referred to in Table 1 as *First*, *All*, and *Constrained*, respectively.

The goodness values computed according to the asymmetric model generally fit the data better than those computed according to the symmetric model, as shown in Table 1 (correlations $\geq$ .582 ($df$=7) or .549 ($df$=8) are reliable at $\alpha$=.05, one-tailed; the power is, of course, quite low). The fit is even better than it appears, for the following reason. There are two sources of error external to the model: either a flaw in the descriptive study produced a distorted set of features from which the goodness was calculated, or a flaw in the design of the experiment distorted subjects' accuracy in selecting the correct picture.

A rather glaring example of the second kind of error occurred in the design of the before-after pictures. Subjects were less accurate in choosing Picture #3 for **copy** than for **repeat**, because on the one hand, Picture #11, with the word "hardcopy" in it, was a distractor (most subjects picked it instead of #3 for **copy**), while on the other hand, the design of Picture #3 made it look somewhat more like a repetition of a preceding line than a copy of an arbitrary line. Because of this, I have excluded it from the rest of the analysis (and as shown in Table 1, the predictive accuracy of the model increases somewhat as a result).

On the whole, then, the assessments of goodness accounted for roughly half of the variance in subjects' accuracy. The one case where this did not occur was in the constrained-choice condition for Nameset 3, where subjects were much more accurate than the goodness of the names predicted. While there was no increase in accuracy in this condition for the other two namesets, and the pattern of choices remained the same, for Nameset 3 there was a radical shift in both choices and accuracy. The reason for this is not entirely clear. It might have been a learning effect, except that the order in which the namesets were presented was counterbalanced, and the other two namesets did not show any increase in accuracy in the constrained condition. It thus seems like an effect of the set of names per se. This leads to the general issue of context effects and the context-free nature of the goodness model.

**Table 1. Rank-order Correlations of Goodness with Number of Subjects (of 12) Picking Correct Picture.**

| | Goodness | | Number Correct | | |
|---|---|---|---|---|---|
| Nameset 1 | Sym. | Asym. | First | All | Constrained |
| Append | .667 | .750 | 5 | 5 | 5 |
| Change | .333 | .400 | 3 | 4 | 0 |
| Copy | .444 | .533 | 3 | 4 | 4 |
| Delete | 1.000 | 1.000 | 8 | 9 | 12 |
| Insert | .750 | .857 | 8 | 8 | 6 |
| Move | .667 | .750 | 8 | 8 | 7 |
| Print | .400 | .571 | 1 | 2 | 2 |
| Quit | .667 | .727 | 9 | 9 | 9 |
| Substitute | .571 | .667 | 0 | 4 | 0 |
| Write | .444 | .571 | 0 | 1 | 1 |
| Correlation with Sym. Goodness: | | | .708 | .834 | .821 |
| w/o Copy: | | | .701 | .824 | .826 |
| Correlation with Asym. Goodness: | | | .637 | .755 | .771 |
| w/o Copy: | | | .657 | .779 | .793 |

| | Goodness | | Number Correct | | |
|---|---|---|---|---|---|
| Nameset 2 | Sym. | Asym. | First | All | Constrained |
| Append | .667 | .800 | 5 | 6 | 5 |
| Change | .500 | .400 | 3 | 5 | 2 |
| Copy | .000 | .533 | 3 | 4 | 4 |
| Erase | .800 | .727 | 10 | 10 | 11 |
| Insert | 1.000 | .857 | 7 | 7 | 7 |
| Transfer | .250 | .471 | 4 | 4 | 4 |
| Display | .667 | .667 | 8 | 8 | 5 |
| Leave | .500 | .571 | 5 | 6 | 6 |
| Substitute | .500 | .400 | 3 | 6 | 3 |
| Store | .333 | .615 | 8 | 8 | 8 |
| Correlation with Sym. Goodness: | | | .571 | .674 | .464 |
| w/o Copy: | | | .466 | .552 | .402 |
| Correlation with Asym. Goodness: | | | .756 | .625 | .777 |
| w/o Copy: | | | .712 | .590 | .697 |

| | Goodness | | Number Correct | | |
|---|---|---|---|---|---|
| Nameset 3 | Sym. | Asym. | First | All | Constrained |
| Add | .667 | .750 | 6 | 9 | 8 |
| Change | .500 | .500 | 5 | 8 | 3 |
| Repeat | .400 | .400 | 9 | 10 | 10 |
| Delete | .571 | .727 | 9 | 10 | 11 |
| Insert | .750 | .857 | 7 | 8 | 8 |
| Move | .444 | .533 | 2 | 5 | 8 |
| Display | .500 | .667 | 7 | 9 | 7 |
| Quit | .571 | .667 | 7 | 8 | 8 |
| Replace | .400 | .444 | 3 | 4 | 2 |
| Store | .333 | .400 | 6 | 8 | 9 |
| Correlation with Sym. Goodness: | | | .299 | .251 | .009 |
| w/o Repeat: | | | .592 | .525 | .206 |
| Correlation with Asym. Goodness: | | | .236 | .203 | .016 |
| w/o Repeat: | | | .607 | .571 | .284 |

It can be seen from the definition of goodness given above that the measure of the goodness of a name is independent of the set of names of which it is a part. This lack of context-sensitivity can lead to prediction errors in two ways: first, names which are fairly similar to a large number of other names in the set, or which are extremely similar to just one other name, could be picked much less accurately than their goodness might suggest. A good example of this is the name *change*, which occurred in each of the three namesets, and was in each case confused with its virtual synonym *substitute* or *replace*. (Note that the accuracy for these names increases dramatically in the Correct-All measure, since the pair were almost always picked together as first and second choices.) Second, a name which is fairly dissimilar to the rest of the names in the set could be picked much more accurately than its goodness might suggest (a similar analysis is suggested by Black and Moran). A good example here is that of the name *store* (*write* might have also shown the effect, except that subjects picked its distractor picture #12 almost all the time).

Inspection of the deviations from predicted accuracy reveals almost all of them to be explainable in terms of such context effects. Presumably then a context-sensitive model would account for these. A forthcoming report will investigate this in more detail.

## CONCLUSIONS.

As a first-order approximation of the goodness of command names, the context-free featural similarity model presented here is fairly successful. An extension of it would probably be even more accurate. What significance does this have, both for designing interfaces, and for other research dealing with different aspects of command languages?

For designers, there is the possibility of using this model to evaluate potential candidates for command names without building an implementation, and before the choices are frozen into a product's design. In order for this to be most useful, however, a way must be found to do these goodness calculations without consulting large numbers of people. It may well be possible to construct features for names and commands by some analytic method based on consulting dictionary definitions or a thesaurus. The goodness algorithm, and context-sensitive extensions of it, are easily implemented on a computer, and designers could evaluate several namesets and their variations in a short time.

The relationship of this research to other work is interesting, but the details have yet to be spelled out. Black and Moran, using different stimuli and methodology, have identified factors influencing learning

and remembering of command names, that is, factors making a name good in the inverse sense of that suggested in this paper. Are these two kinds of factors similar, orthogonal, or conflicting? They are probably correlated, although one can think of situations where they might be independent, for example, synonymous high- and low-frequency words might be equally good (i.e., suggestive) in the sense discussed here, but unequal in goodness (i.e., memorability) in the sense used by Black and Moran. Presumably the optimal name is one which maximizes both suggestiveness and memorability. Further research refining the two approaches should lead to a better understanding of how they are related.

## REFERENCES


1. Barnard, P. J., Hammond, N. V., Morton, J., Long, J. B., and Clark, I. A. Consistency and compatibility in human–computer dialogue. *Int. J. of Man-Machine Studies*, 1981, *15*, 87-134.
2. Black, J., and Moran, T. Learning and Remembering Command Names. *Proceedings of the Conference on Human Factors in Computer Systems.* Gaithersburg, Maryland. March, 1982.
3. Levin, J., and Moore, J. Dialogue games: meta-communication structures for natural language interaction. *Cognitive Science.* 1: 355–420. 1977.
4. Moran, T. P. The Command Language Grammar: a representation for the user interface of interactive computer systems. *Int. J. of Man-Machine Studies*, 1981, *15*, 3-50.
5. Tversky, A. Features of similarity. *Psychological Review.* 84: 327–352. 1977.
6. Tversky, A. Studies in similarity. in E. Rosch and B. Lloyd, eds., *Cognition and Categorization.* Hillsdale, N.J.: Lawrence Erlbaum. 1979.