

- Norman, D. (1988). *The Psychology of Everyday Things*. Basic Books, New York.
- Palmer, S. (1977). Hierarchical structure in perceptual representation. *Cognitive Psychology*, 9:441–474.
- Palmer, S. (1978). Structural aspects of visual similarity. *Memory & Cognition*, 6(2):91–97.
- Regier, T. (1991). Learning spatial concepts using a partially-structured connectionist architecture. Technical Report TR-91-050, International Computer Science Institute, Berkeley, California.
- Schyns, P. and Murphy, G. (1994). The ontogeny of part representation in object concepts. In Medin, D., editor, *The Psychology of Learning and Motivation*. Academic Press. In press.
- Takano, Y. (1989). Perception of rotated forms: A theory of information types. *Cognitive Psychology*, 21:1–59.
- Townsend, J. (1971). Alphabetic confusion: A test of models for individuals. *Perception & Psychophysics*, 9(6):449–454.
- Treisman, A. and Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology*, 12(12):97–136.
- Treisman, A. and Paterson, R. (1984). Emergent features, attention, and object perception. *Journal of Experimental Psychology: Human Perception and Performance*, 10:12–31.

- Hofstadter, D. and McGraw, G. (1994). Chapter 9: Letter Spirit: Esthetic perception and creative play in the rich microcosm of the alphabet. In Hofstadter, D., editor, *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Basic Books, New York. To appear. A similar report is currently available as TR-68 from the Center for Research on Concepts and Cognition.
- Holland, J. H. (1986). Chapter 20: Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems. In Michalski, R., Carbonell, J., and Mitchell, T., editors, *Machine Learning: An Artificial Intelligence Approach*, volume 2, pages 593–623. Morgan Kaufmann Publishers.
- Jackendoff, R. (1994). *Patterns in the Mind: Language and Human Nature*. Basic Books, New York. In Press.
- Kahneman, D. and Miller, D. (1986). Norm Theory: Comparing reality to its alternatives. *Psychological Review*, 93:136–153.
- Keren, G. and Baggen, S. (1981). Recognition models of alphanumeric characters. *Perception & Psychophysics*, 29(3):234–246.
- Kurzweil, R. (1990). *The Age of Intelligent Machines*. MIT Press, Cambridge, MA.
- Lee, M. and Oldham, W. (1990). Font recognition by a neural network. *International Journal of Man-Machine Studies*, 33:41–61.
- Mantas, J. (1986). An overview of character recognition methodologies. *Pattern Recognition*, 19(6):425–430.
- Mantas, J. (1987). Methodologies in pattern recognition and image analysis — a brief survey. *Pattern Recognition*, 20(1):1–6.
- McClelland, J. and Rumelhart, D. (1981). An interactive-activation model of context effects in letter perception: Part 1, an account of basic findings. *Psychological Review*, 88(5):375–407.
- McGraw, G. (1992). Letter Spirit: Recognition and creation of letterforms based on fluid concepts. Technical Report 61, Center for Research on Concepts and Cognition, 510 North Fess, Bloomington, IN, 47408.
- McGraw, G. and Drasin, D. (1993). Recognition of Gridletters: Probing the behavior of three competing models. In *Proceedings of the Fifth Midwest AI and Cognitive Science Society Conference*, pages 63–67.
- McGraw, G., Rehling, J., and Goldstone, R. (1994). Letter Perception: Toward a conceptual approach. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*. To appear.
- Medin, D., Goldstone, R., and Gentner, D. (1993). Respects for similarity. *Psychological Review*, 100:254–278.
- Mitchell, M. (1993). *Analogy-making as Perception*. MIT Press/Bradford Books, Cambridge, MA.
- Mori, S., Yamamoto, K., and Yasuda, M. (1984). Research on machine recognition of handprinted characters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(4):386–405.

References

- Blesser, B., Shillman, R., Cox, C., Kuklinski, T., Ventura, J., and Eden, M. (1973). Character recognition based on phenomenological attributes. *Visible Language*, 7(3).
- Cutler, A. (1982). *Slips of the Tongue and Language Production*. Mouton, Berlin.
- Dell, G. (1986). A spreading activation theory of retrieval in sentence production. *Psychological Review*, 93(3).
- Fromkin, V. (1980). *Errors in Linguistic Performance: Slips of the Tongue, Ear, Pen, and Hand*. Academic Press, New York.
- Fukushima, K. (1989). Analysis of the process of visual pattern recognition by the neocognitron. *Neural Networks*, 2:413–420.
- Fukushima, K. (1992). Character recognition with neural networks. *Neurocomputing*, 4:221–233.
- Gaillat, G. and Berthod, M. (1979). Panorama des techniques d’extraction de traits caractéristiques en lecture optique des caractères. *Revue Technique Thomson-CSF*, 11(4):943–959. In French.
- Gibson, E. (1971). Perceptual learning and the theory of word perception. *Cognitive Psychology*, 2:351–368.
- Gibson, E. J. (1969). *Principles of Perceptual Learning and Development*. Meredith Corporation, New York.
- Gilmore, G., Hersh, H., Caramazza, A., and Griffin, J. (1979). Multidimensional letter similarity derived from recognition errors. *Perception & Psychophysics*, 25(5):425–431.
- Goldstone, R., Medin, D., and Gentner, D. (1991). Relations, attributes, and the non-independence of features in similarity judgments. *Cognitive Psychology*, 23(2):222–264.
- Guyon, I., Poujaud, I., Personnaz, L., and Dreyfus, G. (1989). Comparing different neural network architectures for classifying handwritten digits. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN-89)*, pages II, 127–132, Washington, D.C.
- Hinton, G., Williams, K., and Revow, M. (1992). Adaptive elastic models for hand-printed character recognition. Presented at Cogsci 1991, and posted to Neuroprose.
- Hock, H., Tromley, C., and Polmann, L. (1988). Perceptual units in the acquisition of visual categories. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 14:75–84.
- Hofstadter, D. (1979). *Gödel, Escher, Bach: an Eternal Golden Braid*. Basic Books, New York.
- Hofstadter, D. (1985). *Metamagical Themas: Questing for the Essence of Mind and Pattern*. Basic Books, New York. See especially Chapters 10, 12, 13, 23, 24, and 26.
- Hofstadter, D. (1987). Introduction to the Letter Spirit project and to the idea of “gridfonts”. Technical Report 17, Center for Research on Concepts and Cognition, 510 N. Fess, Bloomington, IN, 47408.

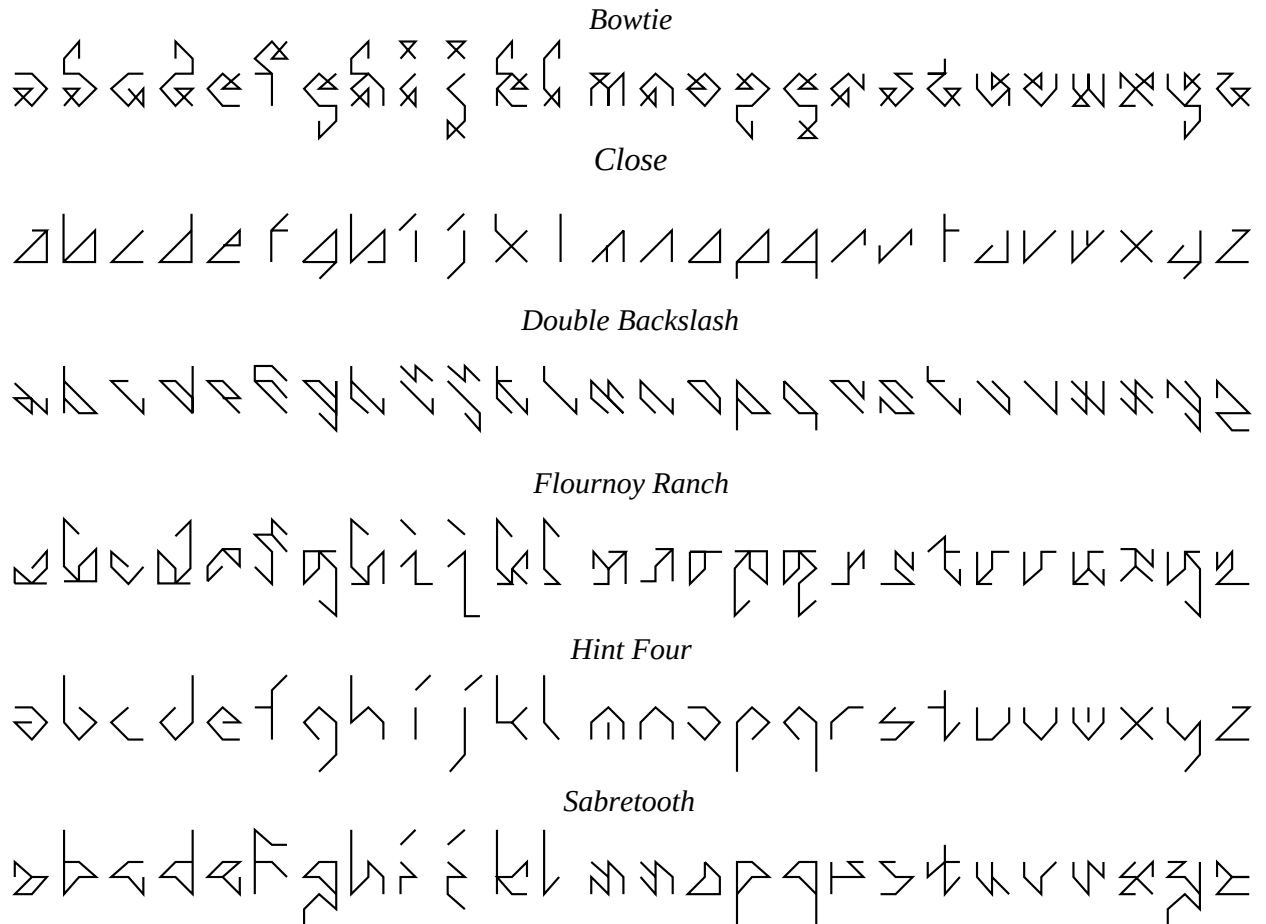


Figure 20: The six gridfonts that we used in the experiment, making up the data set **FONTS**.

[illegible]

Figure 19: The **NORMALS** data set. Each of these letters is meant to be a strong member of its intended category. The first five tokens in each letter category made up the **TRAIN** data set.

Acknowledgements

Douglas Hofstadter supplied much of the motivation for this research. His ideas about cognition pervade this article. The authors also wish to thank Doug Blank, Mike Gasser, Terry Jones, David Leake, Jim Marshall, Lisa Meeden, Bob Port, and Achille Varzi for their helpful comments and criticism.

transformation were rated by people as more similar than patterns related by the first, indicating that similarity is a function of higher-order perceptual units rather than of elementary dot locations.

This result, together with those of Palmer (1977; 1978), discussed in Section 1.1, suggests that higher-order perceptual units play a key role in categorization. Although Palmer’s shapes were somewhat letter-like in appearance (and were made on a grid similar to ours), they were not letters. A natural, although controversial, question naturally follows: Are letters decomposed in a similar fashion into higher order perceptual units, or are they somehow “unitized”? This question is answered in favor of higher-order parts both by our experiments and by those of Treisman and Gelade (1980), discussed above.

Schyns and Murphy (1994) also argue for the utility of segmentation into complex parts. For them, a necessary aspect of such parts is their functional nature. Their work shows that parts are not perceptual primitives, as Treisman would argue (*e.g.*, complex line segments, edges, etc.); rather, they are functional units that have been learned to be important in categorizing and representing the object in question. These units are typically defined in terms of several primitive features. Our theory of roles fits these results perfectly, and argues that functional parts are also useful in letter perception.

Our work suggests that the higher-order pieces into which letters are segmented are key to the categorization process. Roles provide top-down influence on the segmentation of letterforms into such higher-order parts. Our experiments show that human letter-recognition data, especially when the stimuli include non-standard letterforms, can be clearly and easily explained in terms of roles, while other theories are far less powerful in terms of explanation and prediction.

7.4 Sketch of a process model

Our results point out fundamental shortcomings in approaches to letter recognition that do not deal with letters as realizations of roles in the same way as people do. Establishing the importance of roles as units of representation is an empirical step that confirms the architecture of Letter Spirit, our developing computer model of letter recognition (and design). Encouraged by these results, we are proceeding with the implementation of this architecture. Among the critical portions of the recognition process in our model are these: building preliminary parts in a bottom-up fashion from low-level data, adjusting parts under top-down influence from roles, noting norm violations that result from filling roles with particular parts, evaluating prospective filled role-sets as matches of different letter categories, tracking the strengths of activation of competing categories, and finding new parsings if primary attempts at recognition result in only weak categorization. The model we are building, which is explained in some detail in (Hofstadter and McGraw, 1994), has close ties to the Copycat model of Mitchell (1993) — including stochastic parallel processing done by micro-agents that interact with a malleable conceptual network.

Results from the experiments reported above serve to justify the model building in several ways. The results of Palmer (1977) and our follow-up experiment on parsings suggest some guidelines for parsing letterforms into parts. The discovery that humans consider more than one category when presented with ambiguous letterforms and that humans can respond to the exact same stimulus presented at different times with two or more different answers suggests that the overall control principle be a nondeterministic one allowing a number of answers to be considered in parallel. The various tasks in each possible matching, including parsing into parts and filling roles with parts, probably also happen in parallel. We feel that the central finding of this inquiry is that mimicking the human process of letter recognition is an essential — rather than simply an attractive — ingredient of a successful approach to letter recognition.

7.3 Psychological theories of letter perception

7.3.1 Feature lists and mathematical models

There are many different psychological theories of letter perception. Some of the least plausible are based on mathematical methods and have much in common with the OCR approaches addressed above (Townsend, 1971; Gilmore et al., 1979). Keren and Baggen (1981) show why low-level geometric approaches like Townsend's are not psychologically substantive. Although models based on Luce's Choice or multidimensional scaling can often do the curve-fitting required in order to fit human perceptual data (at least of some limited sorts), they tell us little about the psychological processing involved in human letter perception.

Instead, Keren and Baggen (1981) propose a feature-analytic model based on a feature list very similar to Gibson's (1969; 1971) (although their features are arguably more global in nature). Though their model works quite well on the data set they developed, they readily admit that it cannot handle more than one typeface. This points out a major problem associated with the use of feature lists. Tailored feature lists will apply only exceptionally to more than one set of stimuli. They are certainly not general enough to handle multiple typefaces, especially when the styles involved go out towards the fringes of letter-categories, as is often the case in display faces for advertising. Feature models developed to date use features that are too limited. It is highly unlikely that developing a set of general features for letter recognition will ever be possible (Hofstadter, 1985). It is certainly the case that feature-based representations of complex stimuli like letters are more easily and naturally modeled by richer representational systems.

Treisman and Gelade (1980) argue that "local elements or parts of shapes function as separable features which must be integrated by focused attention." They then go on to say that this notion of separable features applies to letters as well. One of their experiments involved asking subjects to search for a target 'R' in a background of 'P's and 'Q's or a background of 'P's and 'B's, the idea being that if the letters are being decomposed into parts and then re-integrated, there will be illusory conjunctions in the first condition and not the second, causing the first condition to have a linear time increase with display size. Their results suggest that feature-integration theory applies to letters as well as to complex shapes, and that letters are indeed broken into higher-order parts and re-integrated. We agree that letterforms are made up of parts that together define a structural whole, but we go further and argue that the parts themselves are higher-level structures and not single, complex features as Treisman and Gelade state.

Problems with feature lists can be summed up as follows. Feature lists: 1) lack the ability to easily expand, since features are not built of building blocks, and are by definition non-decomposable; 2) cannot represent a writing system with a potentially infinite number of features, as letterforms, especially advertising faces, seem to require; 3) are not efficient, since implausibly large numbers of features are often required; and 4) cannot express the intuitively clear relationships between different features. Considered in this light, the benefits of higher-level representations certainly outweigh the complications they may add to processing in a model.

7.3.2 Higher-level relational structures

Hock et al. (1988) show that similarity and categorization judgments are based on parts and relational configurations in a pattern, rather than on elementary constituents of a pattern (*e.g.*, quanta or dots). Their patterns consisted of 8 dots randomly placed on a two-dimensional space. Subjects judged the similarity of a dot pattern to two transformations of the dot pattern. In one type of transformation, a single dot was moved to a new area of the space. In the other type of transformation, three neighboring dots were moved as a unit to a new area. Patterns related by the second

is required (Jackendoff, 1994; Hofstadter, 1985). The simple models reported here contrast sharply with the human case. It is clear that more sophisticated models are required. We suggest that future models should take the conceptual roles inside letters into account.

7.2 Machine-based approaches

(Mantas, 1987; Mantas, 1986; Mori et al., 1984; Gaillat and Berthod, 1979) all provide overviews of methodologies in character recognition from the engineering standpoint. These studies, which focus on OCR, show that for the most part, engineering research concentrates on statistical and mathematical methods (*e.g.*, nearest neighbor, Bayes rule, hierarchical clustering) applied to low-level local features of letterforms. Of course, most engineering solutions are not meant to provide cognitively plausible models of letter recognition, but it is clear that the limitations of such models are a direct consequence of their non-human approach to recognition. There is much the field can learn from cognitive psychology.

Although some OCR systems are commercially available today, a survey of the literature shows that the problem of OCR has not been satisfactorily solved. A system that can recognize letters in many typefaces that it has never seen (or been trained on) has yet to be developed. In (Kurzweil, 1990, page 241), OCR pioneer Raymond Kurzweil states, “While machines exist that can accurately recognize many type styles in common usage, no machine can successfully deal with the level of abstraction required by ... ornamental forms.”

Gaillat and Berthod (1979) present a taxonomy of classical OCR systems that includes the following broad classification of methods:

- Point-to-point comparison
- Global transformations (*e.g.*, Fourier transforms or the method of moments)
- Local features and curvatures
- Structural decomposition into parts

This list still seems adequate to classify most systems that have been developed in the fifteen years since their survey was made. The results of Section 6 show the limitations built into the first three of these approaches, and recommend heavily the fourth.

Connectionist models, about which claims of cognitive plausibility are often made, have also been applied to letter recognition, but with only limited success. The interactive-activation reading model of McClelland and Rumelhart (1981) makes use of a geometric feature-based recognizer. Its letter-recognition capability is limited to uppercase letters of one uniform typeface. Indeed, each letter “concept” learned in the model is simply a specific, fixed shape. Other network models, also based on integration of low-level geometric features, exist in the literature (Hinton et al., 1992; Guyon et al., 1989; Lee and Oldham, 1990; Fukushima, 1992). These models have not been shown to be capable of recognizing ornamental or eccentric or fanciful letters such as those found in **FONTS**.

Very little work on machine-based recognition systems has considered high-level features, let alone collections or groups of such features (the building blocks of roles) (Mori et al., 1984; Gaillat and Berthod, 1979). This is probably because OCR research has had little to do with cognitive psychology. We are convinced that without considering the human recognition system and attempting to model it, letter recognition by machines will remain inferior to that by people.

local features	
Error	Error Rate
m↔n	16.10
u↔w	7.90
c↔e	6.20
b↔h	5.80
o↔u	4.25
v↔w	4.00
u↔v	3.85
e↔o	3.85
n↔o	3.60
n↔r	3.40

Table 13: The top ten bidirectional category errors from the local-features prediction matrix.

proto-role slippage		
Error	Error Rate	Human Rank
u↔v	26.60	1
g↔q	13.60	2
i↔j	12.70	4
f↔t	12.60	5
i↔l	11.95	3
e↔p	5.90	17
a↔d	5.90	11
n↔r	5.25	8
c↔r	5.10	9
k↔x	2.30	21

Table 14: The top ten bidirectional category errors from the proto-role-slippage prediction matrix. The ranking of each error in terms of human data is listed for comparison purposes.

model. It is interesting to compare this list with the human errors shown in Table 4. The proto-role-slippage model correctly predicts the top five human errors (although in a slightly different order), and includes seven entries in the top ten. Such predictive power is noteworthy indeed.

It is also worth mentioning the performance of our proto-role-slippage model in terms of a comparison made between a global-feature-based model and Luce’s Choice model. Keren and Baggen (1981) compare the choice model and the global-features model on the data obtained by Gilmore et al. (1979) for uppercase letters. They found that the choice model had a high degree of fit with the entire error matrix ($r = 0.995$) but that it was due to curve-fitting in a 350-parameter model, which gave little insight into the actual psychological processes. They offer their own model, based on global features (as described above) and obtain a competent degree of fit ($r = 0.936$) in a model that weights the importance of a list of global features, and that has only 29 parameters. In comparison, our proto-role-slippage model explains psychological mechanisms in a far more vivid manner, and correlates with our entire matrix (including the diagonal also, now) extremely well ($r = 0.995$). Our approximation of intercategory distance is accomplished with 5 parameters. The superiority of proto-role slippage as a method of determining intercategory distances strongly recommends roles as a crucial level of representation in letter-recognition systems.

7 Discussion

7.1 What is missing in the Dumrec and Netrec models?

The main weakness of the Dumrec and Netrec models is a lack of flexible higher-order representations in the recognition process. There is obviously more to human letter recognition than the noticing of simple geometric features (as in Dumrec and Netrec). Although the enhanced Dumrec models make use of slightly more abstract properties, they still rely too heavily on low-level geometric features (as is apparent from Table 12). People are much better at recognition precisely because their representations of letterforms are more abstract. There are certainly limits to the amounts of slippage that people’s representations can accept, but some ability to adjust flexibly under pressure

Intercategory distance calculation in the proto-role-slippage model			
cats	formula	calculation	distance
$f \leftrightarrow t$	$(\text{diff in staff orientation})^2 + \Sigma (\text{diff staff curvature}) + \Sigma (\text{diff arm direction})$	$0^2 + (1 + 1) + (0 + 0)$	2
$a \leftrightarrow g$	$(\text{diff in attachment pt})^2 + (\text{diff in stem curvature})$	$2^2 + 1$	5
$a \leftrightarrow b$	$(\text{diff in attachment pt})^2 + (\text{diff in stem curvature})$	$2^2 + 3$	7
$j \leftrightarrow l$	$(\text{diff in staff orientation})^2 + \Sigma (\text{diff staff curvature}) + (\text{dot diff})$	$0^2 + 2 + 2$	4
$y \leftrightarrow o$	dissimilar types	constant factor of 10	10

Table 11: Intercategory distance calculation examples taken from the proto-role slippage model.

6.3 Results from the theoretical error-prediction matrices

Pearson r -values			
	Syntax	Local Features	Proto-role Slippage
Syntax	1.00		
Feature	0.81	1.00	
P-role def.	0.32	0.25	1.00
Netrec	0.54	0.40	0.30
Dumrec-S	0.68	0.62	0.33
Dumrec-NT	0.74	0.67	0.42
Dumrec	0.60	0.68	0.41
People	0.37	0.32	0.88

Table 12: Correlations between prediction matrices, computer models, and people. This table sheds light on the kind of processing done by the models and by people.

Table 12 shows the Pearson r -values comparing all of our models, and humans, with the three theoretical models. These correlations take into account only off-diagonal confusions.¹⁸ It is clear that the programs rely heavily upon simple syntax and local features in arriving at their decisions, with up to half of the variance of the computer models coming from these low-level factors. Humans, on the other hand, hardly use this information at all. Much of the variance in human data that can be explained by syntax or local features is likely due to the correspondence between those simpler models and the proto-role-slippage model. This argument can be concisely stated in terms of the role hypothesis: roles affect the kinds of parts that a letter can have by providing norms for such parts; specific parts of a letterform are syntactic in nature; thus roles provide top-down constraints on the syntactic level of letters, and their effects on the syntactic level can be easily seen.

The comparison between the proto-role-slippage model and the actual human numbers shows a very strong correlation ($r = 0.88$), demonstrating that this model captures some aspects of human letter representation fairly well. On the other hand, the behavior of the computer models is predicted only to a small extent by the proto-role-slippage model. It is worth noting that the most outrageous Dumrec and Netrec errors — exhibited in Figures 17 and 18 — are those in which the token has a strong resemblance to the incorrectly guessed category if viewed on the syntactic or local-feature level, but in which the mutation at the level of proto-roles would have to be huge, and so such errors are never made by humans.

Table 14 shows the top ten bidirectional category errors predicted by the proto-role-slippage

¹⁸We omitted the diagonal so that the correlations would not be overwhelmed by the strength of the correct category vector.

6.2.4 Proto-role slippage

Finally, we created a matrix of distances between categories that serves as a very rough first approximation to our role model. We posited that the distance between two categories would be a function of the number of discrete mutations, or *slippages*, needed to transform the representation of a category into another category’s representation.¹⁶ Each letter category can be slipped into others with differing amounts of ease. “Slippage” involves allowing certain descriptions in a mental representation to “slip”, or be replaced by closely related descriptions according to conceptual pressures being brought to bear by the situation (Hofstadter, 1979; Mitchell, 1993). Measuring the slippage distance between categories provides a formal way of calculating intercategory distance.

To do this, we first had to encode each letter category in terms of its *proto-roles* (a simple-minded and rough first approximation to roles). In our implementation there were only two very crude classes of proto-roles: closures and curves of various types. Table 10 shows the representations (translated into English from Scheme) of nine letter categories in terms of proto-roles.

Proto-role sets	
a	closure with sharp left hook attached at NE
b	closure with straight post attached at NW
f	vertical cross with top hooking gently right, straight bottom, arms pointing W and E
g	closure with right hook attached at SE
j	vertical staff hooking right below and having a dot above
l	vertical staff
o	closure
t	vertical cross hooking gently left below, straight top, arms pointing W and E
y	NE-pointing staff with arm pointing NW protruding from middle

Table 10: Proto-role sets defining nine of the twenty-six letter categories. Even with very crude representations such as these, the proto-role slippage model proved powerful.

Slippages between letter categories were measured by the weighted sum of: changes in degree of curvature or in orientation of a proto-role, the addition or deletion of a proto-role, and the change in global orientation of the figure raised to a constant power (2).¹⁷ Orientation was measured in units of 45 degrees. Each type of slippage was assigned a weight according to its estimated impact: changes in degree of curvature or in local orientation (1), addition or deletion of a proto-role (2 or 5, depending on object size), and changes in global orientation (1). The four coefficients, plus the exponent applied to global orientation, are the five parameters that were adjusted to fit the slippage model to human data. The distance between any two categories was the minimum sum of slippage weights required to convert the description of the one category into the description of the other. Table 11 shows five examples of intercategory distance calculation in the proto-role-slippage model.

The proto-role letter-category representations constitute an extremely crude approximation to a detailed model of roles, but are sufficient to demonstrate the importance of deformable roles within human letter recognition. Table 14 (further discussed below) shows the top ten bidirectional errors predicted by the proto-role-slippage model.

¹⁶ “Slippability”, a term coined by Hofstadter (1979), is roughly equivalent to the more pervasive psychological term “mutability” used by Kahneman and Miller (1986).

¹⁷ The global orientation was squared because it was clear that the difference between categories due to global orientation grew faster than linearly.

overall percent correct of the matrix to be in accord with that which resulted from a particular data set and particular stimulus-presentation conditions. It does not, however, change the order relation among errors.

6.2 Theories of intercategory proximity

We consider four different theories of intercategory proximity, which use representations based on: syntax, local features, global features, and proto-role slippage. The syntax model simply amounts to noting the presence or absence of quanta in a letterform, and corresponds to the level-0 data used in Dumrec. The local-features model results in a vector recording each of the intersections and ends of letter segments (such as T-junctions and tips) tagged with the features' location. In the global-features model, properties are derived from the entire gestalt of a letterform, such as "nonclosed letter standing on a broad base" (Keren and Baggen, 1981). Finally, we introduce proto-role slippage, which involves considering each letter category as a set of proto-roles. Any letter category can be converted into any other by a combination of proto-role substitutions.

6.2.1 Syntax

Calculation of syntactic distances was achieved by finding distances between the blurred-prototypes described in Section 3.4. Distances between the 56-dimensional blurred-prototype vectors were calculated using the city-block method (also explained in Section 3.4).

6.2.2 Local features

OCR systems have employed detection of local features as a basis for recognition (Guyon et al., 1989; Gaillat and Berthod, 1979). A list of all possible local features on the grid (such as quanta-intersection points) was created and "blurred feature prototypes" were created using the resulting 1,696-dimensional space (in the same manner as with the blurred-prototype model discussed above). Once again, distances between letter category vectors were calculated using the city-block method. The top ten bidirectional category errors predicted by the local-features model are shown in Table 13.

6.2.3 Global features

Psychologists have proposed several feature lists as bases for human perception of various figures, notably uppercase letters or numerals. These features are usually global and boolean in nature, and in this scheme a letter category amounts to a vector of 1's and 0's, each of which corresponds to the presence or non-presence of one of the global features. We sought to create theoretical error matrices for at least one such model, but found that these models were too rigid to apply beyond their original uppercase data sets. The feature lists of (Gibson, 1971) and (Keren and Baggen, 1981) were intended to discriminate between uppercase letters, and fail to discriminate between lowercase letters. That is, for both features sets there are several pairs of lowercase letters that contain exactly the same set of features. No theoretical error-matrix of reasonable quality could be formed on the basis of either feature list (so we did not create one). Although the originators of these lists admit as much, this shows plainly that reasonably short high-level feature lists cannot be constructed that serve to explain recognition of larger sets of stimuli. Lacking generality, they cannot serve as a basis for a theory of categorization.

over a large range of target accuracy values. The data we report reflect tuning to the human accuracy value, as the human data are of primary interest.

Its abstraction tools are not yet powerful enough. More light is shed on the syntactic nature of Dumrec’s processing in the following section.

6 Theoretical error-prediction matrices

The comparisons with human data drawn above show the limitations of the two computer models we have implemented. In particular, they are brittle when exposed to the exotic letterforms of **FONTS**, and often make poor category judgments. While Dumrec and Netrec are two simple but specific computer models of recognition, we can also make comparisons with theoretical error-prediction models. Such theoretical models, although they are not themselves models of the recognition process but solely of intercategory distances, provide an additional way to characterize the behavior of recognition systems — either human or computer — in terms of their representation of categories. We investigated three types of models, using category representations based on either syntactic overlap, low-level features, or role-level similarity.

We suppose that any letter category can be thought of as a vector representing a point in some high-dimensional similarity space. The distance between letter categories in this space is directly determined by a theory of representation of categories. For example, consider the syntax-based “blurred prototypes” (introduced in Section 3.4), in which each category is represented as a 56-dimensional vector — this representation is based on a theory of *syntactic* proximity of categories. There are a variety of different theories that can be used to define these representations. Below we discuss four different theories of intercategory proximity, each of which gives rise to its own predictions of the likelihood of intercategory confusion.

6.1 Generating error-prediction matrices

Given a theoretical method for defining categories, we can generate theoretical predictions of error rates by first computing the distances between all possible pairs of letter categories (resulting in a 26×26 distance matrix), and then transforming this matrix into a 26×26 error-prediction matrix using a power function, as described below. By comparing (via correlation) the resulting error-prediction matrices with those of an actual system (human or machine), we can find out how closely processing in the actual recognition system coheres with the category theory used to build the theoretical matrix.

More technically, our error matrices were generated as follows. Each theoretical model provides a method of representation of categories (all are described in detail below). Using these representations, we compute a distance between every pair of letter categories. Every entry, d_{ij} , in the resulting distance matrix is then transformed using a function well known to psychologists: $a_i e^{-k d_{ij}}$, where a_i sets the total of each row to unity, and k is adjusted to set the overall percent correct (average of entries along the main diagonal) to a particular value. This exponential formula converts each unit increment in intercategory *distance* into a reduction of that intercategory *error* by a constant factor. Greater distances mean rarer errors (hence the negative sign in the exponent). The value of k affects dispersion of values away from the main diagonal. Large values of k will leave the diagonal entries at 1 (since the distance between a category and itself is always zero), but will minimize any errors. Very small values of k (near 0) bring all entries closer to equality. For each theory’s matrix, we tuned k so that the overall percent correct of the matrix (the average of entries along the main diagonal) equals the overall percent correct of the human data.¹⁵ This factor therefore adjusts the

¹⁵For the purposes of the correlations below (shown in Table 12), we set the total percent correct (*i.e.*, accuracy) of each theoretical model to the human value. Experimentation showed that correlation r -values were quite consistent

and $g \leftrightarrow q$) are predicted by the closest blurred-prototype categories shown in Table 3. This implies that to some extent, processing in the nets relies more heavily on syntactic quanta-level features than does Dumrec processing. The $s \leftrightarrow z$ error fits the syntactic-feature theory as well — although it shows that Netrec is doing more than simple overlap of quanta — since gridfont ‘s’s and ‘z’s share similar features at the x-height and the baseline. The $b \leftrightarrow l$ error probably occurs because both ‘b’ and ‘l’ extend into the ascender zone and some of the ‘l’s have diagonal serifs in the central zone.

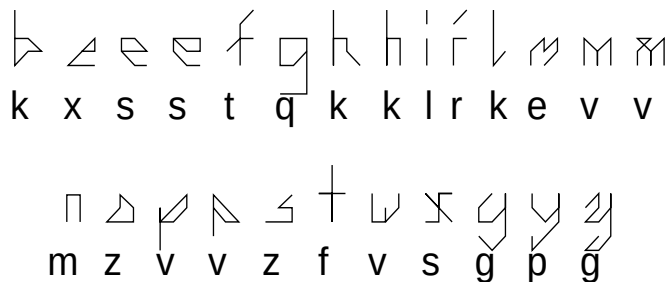


Figure 18: A complete list of Netrec errors that were never made by people. Netrec’s incorrect categorization is shown below each of the tokens.

Figure 18 shows all of the tokens that Netrec misclassified but that people always got right. There are 25 such errors, roughly twice as many as Dumrec’s. It is clear from studying the figure that most of the errors are caused by low-level feature mistakes. See especially $e \rightarrow s$, $g \rightarrow q$, $n \rightarrow m$, and $t \rightarrow f$, which are predicted by the closeness of letter categories in syntactic space as defined by blurred-prototypes (see Table 3). These errors reinforce the theory that the network is paying close attention to low-level geometric features near the quanta level instead of to higher-order parts. Some of the other errors are clearly based on collections of particular simple features or sets of quanta. Consider, for instance, the $i \rightarrow l$ error, the $i \rightarrow r$ error, and the $b \rightarrow k$ error. In general, errors made by the network seem much less reasonable than people’s errors (see especially the $m \rightarrow e$ error). Taken together with the correlation information, these data imply that simple feedforward networks of the kind we tested are not good cognitive models of letter perception. They are not able to learn associations that are general enough to correctly classify new letterforms. This is further corroborated by theoretical results introduced in Section 6.

5.3 Lesioned Dumrec

The general trend of the correlation results in Table 7 shows that the full Dumrec model correlates better with human data than its lesioned counterparts do. The slightly more abstract features included in Dumrec clearly improve performance. This is also evident in Figure 16, which shows the models progressing towards human performance as abstraction is added. However, improvement in performance is not evident for the **FONTS** data set. Curiously, in fact, a smaller opposite effect is seen in **FONTS**, with the lesioned models outperforming the full model. However, the importance of this anomalous effect is tempered by the generally poor correlations found on the **FONTS** data set.

The basic lesson is that abstract features of the kind added to Dumrec seem to help, but this type of feature-based model is still not powerful enough to handle eccentric letterforms. This difference in processing was probably magnified by the fact that the model was tuned using normal shapes. Dumrec is still too syntactic — too *shape-based* — to be able to deal well with category outliers.

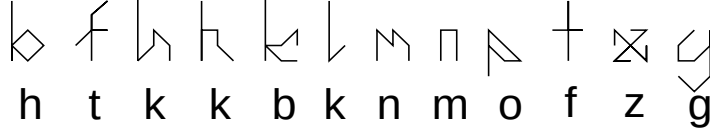


Figure 17: A complete listing of Dumrec errors that were never made by people. Dumrec’s incorrect categorization is shown below each of the tokens.

Error	Dumrec	Humans
m↔n	21.90	2.70
u↔w	19.05	0.85
v↔w	17.40	4.75
e↔z	15.05	1.65
a↔z	14.65	0.85
f↔t	13.10	5.75
h↔k	12.70	4.50
c↔e	12.25	1.20
g↔q	11.60	10.95
g↔y	11.45	0.40
Human error sum	33.6	

Table 8: The top ten bidirectional category errors (by percentage) made by Dumrec. Human error rates (also percentages) are listed for comparison purposes. The human error sum provides for a simple comparison between Dumrec and Netrec.

Error	Netrec	Humans
c↔o	13.65	0.65
e↔s	11.55	0.20
s↔z	10.60	1.25
v↔w	10.20	4.75
f↔t	9.00	5.75
u↔w	8.60	0.85
o↔u	8.60	0.70
a↔s	8.45	0.30
g↔q	8.40	10.95
b↔l	8.35	1.90
Human error sum	27.3	

Table 9: The top ten bidirectional category errors (by percentage) made by Netrec. Human error rates are listed for comparison purposes.

that seem to be based on syntactic processing, but it does not make as many of these errors as Netrec does. The five worst errors are $k \rightarrow b$, $l \rightarrow k$, $n \rightarrow m$, $p \rightarrow o$, and $x \rightarrow z$. The first two errors demonstrate weaknesses in the Dumrec code: the $k \rightarrow b$ error occurred because Dumrec’s closure-rating system is not sophisticated enough (in this case, the ‘b’ that won barely beat out a ‘k’), and the $l \rightarrow k$ error happened because all of the ‘l’ prototypes happen to have their ascender in the *middle* of the grid, whereas the ascender of this particular ‘l’ was placed, without thinking, against the *left edge* of the grid. Since Dumrec is very sensitive to placement, it saw this letter as being very distant from all ‘l’s, and much closer to letters like ‘b’, ‘k’, and ‘h’. The remaining three of the five worst errors are caused by the weakness of the tips calculations: the skinny ‘n’ matches ‘m’ prototypes’ tips better than it does ‘n’ prototypes; the ‘x’s tips are all too tricky for Dumrec (since they are not in or near a standard location); and the short stem on the ‘p’ causes a tip problem. In general, the abstract properties that Dumrec makes use of seem to help. However, their level of abstraction clearly stops short of what is needed. The brittleness of the Dumrec program is clearly brought out by these errors.

5.2.2 Netrec

Table 9 shows the top ten bidirectional category errors made by Netrec. All but three of the errors ($f \rightarrow t$, $g \rightarrow q$, and $v \leftrightarrow w$) are rarely made by humans. Four of the top ten errors ($c \leftrightarrow o$, $e \leftrightarrow s$, $u \leftrightarrow w$,

only on the most normal of typefaces. If a computer program is to be taken seriously as a cognitive model of human letter perception, it must also account for recognition data on more spirited typefaces. There is a brittleness often found in standard artificial-intelligence programs that causes them to break when exposed to slightly richer input than usual (Holland, 1986). Dumrec’s brittleness is evident in the dramatic drop-off between its performances on **NORMALS** and on **FONTS**. People’s performance degrades by 19.8% while Dumrec’s performance falls 36.6%.

The most surprising result in Table 7 is that the lesioned Dumrec models actually outperform the full model on the **FONTS** data set. All the models generally perform poorly, however. This suggests that even the strongest Dumrec model is a weak model of human letter perception (as are most other models in the literature). This is further discussed in Section 5.3.

Also of note is the fact that the correlations of the Netrec networks with people place them clearly ahead of Dumrec-S (at least on the “normal” data sets). This implies that such networks are doing something more than simple quanta-level processing, but are not able to find and exploit some of the more abstract features used in Dumrec. Both of these suspicions are confirmed by the analysis given in Section 6.

Principal-component analysis is a tool that allows us to globally examine the set of programs and people. By considering each program (and the composite human performance) to be a point in 676-dimensional space determined by the values in its error matrix, we can locate the principal dimensions that separate the programs from one another and from people. Points for the programs and for people are plotted according to their values along the first two principal components in Figure 16. The most significant dimension seems to measure something similar to overall percent correct. The second dimension simply distinguishes between Netrec, the Dumrec group, and people, suggesting perhaps fundamental differences. Higher dimensions were not easily interpretable. As can be seen, Dumrec outperforms all other models on this scale as well, and as Dumrec becomes more abstract, it moves towards people along component 1.

5.2 Error-making by the programs

5.2.1 Dumrec

Table 8 shows the ten most common bidirectional category errors made by Dumrec (*i.e.*, the percentage of the time that tokens in either category are misclassified into the other). Dumrec’s top errors have more in common with human errors than do Netrec’s (shown in Table 9). One simple way to see this is to consider the sums of the human errors that correspond to the models. The model whose errors correspond more strongly to those of humans will have a higher sum. This is the case for Dumrec.

Four of Dumrec’s top errors ($f \leftrightarrow t$, $g \leftrightarrow q$, $h \leftrightarrow k$, and $v \leftrightarrow w$) can be found among the top ten human errors (see Table 4). Two of the remaining errors ($n \leftrightarrow m$ and $u \leftrightarrow w$) are predicted by syntactic proximity as defined by the blurred-prototype model (see Table 3). The errors $a \leftrightarrow z$, $c \leftrightarrow e$, $e \leftrightarrow z$, and $g \leftrightarrow y$, however, are not.

Dumrec’s performance is far closer to human performance than is Netrec’s, as is shown in part by a comparison of Tables 8 and 9. These numbers are supported by the inter-model correlations of Table 7. Together, these results suggest that some properties such as those used by Dumrec play a role in human letter perception.

Figure 17 shows all of the tokens that Dumrec misclassified but that people always classified correctly. There are 12 such errors. Although three of the “token errors” are predicted by the naive blurred-prototype model (Table 3), these errors seem generally more reasonable than the Netrec errors (discussed in the next section). It is certainly the case that Dumrec makes *some* dumb errors

ALLbutTRAIN					
	humans	Netrec	Dumrec-S	Dumrec-NT	Dumrec
humans	1				
Netrec	.63	1			
Dumrec-S	.50	.56	1		
Dumrec-NT	.66	.68	.58	1	
Dumrec	.74	.71	.54	.77	1

NORMALS minus TRAIN					
	humans	Netrec	Dumrec-S	Dumrec-NT	Dumrec
humans	1				
Netrec	.70	1			
Dumrec-S	.56	.68	1		
Dumrec-NT	.83	.85	.70	1	
Dumrec	.92	.83	.62	.94	1

FONT					
	humans	Netrec	Dumrec-S	Dumrec-NT	Dumrec
humans	1				
Netrec	.33	1			
Dumrec-S	.43	.45	1		
Dumrec-NT	.38	.36	.43	1	
Dumrec	.22	.21	.43	.49	1

Table 7: Correlations of the confusion matrices of each model (including all error-making information) on three data sets.

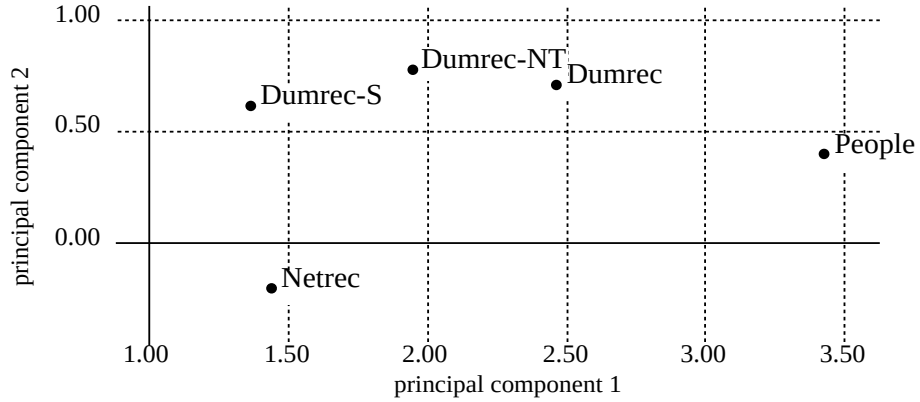


Figure 16: All models compared on the first two principal components of the 676-dimensional space made by turning the 26-by-26 category confusion matrix generated on **AllbutTRAIN** into a vector for each model.

quantum information that we provided directly to our programs. Various parameters of stimulus presentation irrelevant to high-level activity could affect human error rates. We attempted to tune the amount of noise in the stimuli (by regulating the level of visibility of the surrounding grid) so that human and computer error rates could be reasonably compared. The differences in relative proportions of errors made should, we reasoned, be due primarily to fundamental differences in representations at *higher* levels.

Stimulus	humans	Dumrec models			Netrec
		S	NT	Full	
ALLbutTRAIN	81.0	55.9	62.7	69.6	59.8
NORMALS – TRAIN	88.5	64.5	74.9	83.4	71.0
FONTS	68.7	41.7	42.3	46.8	41.0
<i>bowtie</i>	52.1	38.5	53.8	53.8	38.5
<i>double backslash</i>	66.2	34.6	42.3	46.2	34.6
<i>flournoy ranch</i>	57.4	19.2	15.4	19.2	19.2
<i>hint four</i>	85.3	80.8	65.4	76.9	73.1
<i>close</i>	74.8	34.6	26.9	26.9	26.9
<i>sabretooth</i>	76.7	42.3	50.0	57.7	53.8

Table 6: The people vs. the programs. Recognition performance of the models and people on several data subsets. All numbers recorded here are correctness percentages.

Table 6 shows accuracy values for all models and humans on several data subsets. Note that the large data set reported here is **ALLbutTRAIN**, not **ALL**. This is because the programs were trained on **TRAIN**, and testing the models on the training letters is of no value at all. In particular, Dumrec immediately recognizes its prototype tokens and categorizes them correctly every time. Netrec’s performance on the training set is also very good, although not perfect. Since we are interested primarily in the generalization performance of the models (which sheds more light on the representations the models use), we chose not to count performance on **TRAIN**. As in the human case, we tested the models only on letters they had never seen before. The numbers reported below for the Netrec model are an average of the performance of twenty different networks, each fully trained on **TRAIN**.¹⁴

The best overall performance (in terms of accuracy) is that of the complete Dumrec model. The Dumrec model consistently outperforms the lesioned Dumrec models except in two cases — the fonts *hint four* and *close*. The simple connectionist networks fall somewhere between the Dumrec-S and Dumrec-NT, with nothing exceptional about their behavior.

Table 7 shows three correlation matrices for all models and humans on **ALLbutTRAIN**, correlated over entire confusion matrices. Since error-making tends to highlight the type of processing a cognitive system is doing, it is important to take errors into account during correlation. This is one of the reasons we developed the continuous-valued version of Dumrec. Use of the “probability scoring” results in a far richer error-matrix for the Dumrec models.

The difference in correlation between models and humans on **NORMALS** vs. **FONTS** is striking. All of the models correlate much more strongly with people on the simple data set than on the fonts. Dumrec has the highest correlation (0.92) on the **NORMALS** and the lowest on the **FONTS** (0.22). This suggests that models that do reasonably well on conservative characters will not necessarily work well at all on eccentric characters. Most models in the literature to date have been tested only on one or two unadventurous typefaces, and commercially available machines work

¹⁴Each of the twenty networks was initialized with a different set of random weights before training.

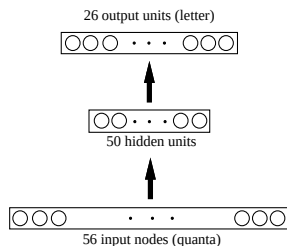


Figure 15: Netrec’s architecture. The solid arrows denote full connectivity.

same as the one provided to Dumrec. The input layer consists of 56 nodes, one for each quantum on the grid. If a quantum is “on” (*i.e.*, it has a line drawn in its position), the quantum’s associated node gets high activation. If it is “off”, its node gets little or no activation. The output layer is simply a vector of 26 nodes, one for each letter of the alphabet. We have experimented with hidden layers consisting of anywhere from 10 to 70 nodes.¹² The network is trained to associate patterns on its input layer with patterns on its output layer, using standard backpropagation of error. We have also trained and tested networks using a richer “feature-based” connectionist approach more similar in nature to those found in (Fukushima, 1989), (Guyon et al., 1989), and (Regier, 1991). Our approach was to train a number of small “subnets” to detect certain features, and then combine these subnets together into a letter-recognizer. Recognition results for such feature networks were not significantly better than or even different from those of regular three-layer nets. For this reason we have chosen not to report them here. See (McGraw and Drasin, 1993) for more information.

One of the open problems in connectionism involves deciding how to pick both a learning rate for backpropagation and the number of hidden units to use. We know of no theoretical results that help guide the choice of these two crucial parameter values. There are some fairly obvious rules of thumb that can be consulted for guidance, but they do not always apply. We opted to determine values for these two parameters experimentally by testing all pairwise combinations of networks with 14 different learning rates (.001, .01, .05, .1, .15, .2, .25, .3, .35, .4, .45, .5, .8, and .99) and 10 different numbers of hidden units (10, 20, 30, 40, 50, 55, 58, 60, 65, and 70) on two different learning tasks. Based on results of experimentation, we decided to use a learning rate of 0.4 and 50 hidden units for the comparison tests reported below.¹³ The backpropagation momentum rate was set to 0.00001.

5 Performance of the models

5.1 General results

Implementation of the computer models was motivated in part by the potential for comparison with the human subjects. Of course, there are important inherent differences between the human and machine conditions, so that comparisons are meaningful only if these differences are taken into consideration. For instance, all of the recognition programs employ simple feedforward processing, so time of performance is always constant, and hence any information regarding timing is meaningless. Another important difference is that in the case of human perception, the pixel-level stimuli must first pass through the low-level portions of the visual system in order to arrive at the medium-level

¹²Some experimentation has also been done with perceptron nets having no hidden layer.

¹³In the course of searching for these values, an interesting result regarding non-convergence with nets of greater than 70 hidden units was found. This result is briefly discussed in (McGraw and Drasin, 1993).

Properties included in the model		
Level-0 only <i>simple</i>	Levels 0-2 <i>no tips</i>	All properties $\approx$ <i>abstract</i>
Dumrec-S	Dumrec-NT	Dumrec

Table 5: The three flavors of Dumrec. “Properties included” corresponds in some sense to lesioning the model.

of the winning token. We also extensively tested a WTA-*category* selection scheme (computed by averaging together the scores of the five prototype tokens in each of the 26 possible categories, and then picking the category with the highest average score), but performance using this scheme was essentially the same as that of models using the WTA-token scheme.

For some of the results reported below, the WTA-token selection scheme was enhanced in order to make Dumrec’s output continuous-valued (and no longer winner-take-all), so that its performance could better be compared with that of humans. To make the program less dependent on winner-take-all selection, we used an exponentially-decreasing scoring function to assign a “probability score” to each letter category based on its prototype tokens’ relative rankings in terms of score. Each of the 26 categories was awarded points: the highest-ranking prototype’s letter category was awarded 0.5 points, the next highest 0.25, then 0.125, and so on, decreasing by half until the number of points awarded became negligible (after 6 applications). Note that the sum of all these scores is almost equal to 1. The exponentially-decreasing function was especially appropriate in this case for two reasons: firstly, it provided a natural way to change the winner-take-all rank ordering into a continuous-valued model, while preserving the winner-take-all ranking; and secondly, because **TRAIN** had 5 prototype letters for each letter category, it lent itself to a scoring function that gave the most weight to only the top few ranks. The resulting “probability scores” for the top categorizations of each mystery letter allow for the construction of a fuller, more realistic error matrix for the Dumrec models.

Although use of the exponentially-decreasing scoring rule certainly affected correlations of the model with human data, the overall effect of changing selection rules was minor.

Altogether we studied three versions of Dumrec, two lesioned and one non-lesioned: Dumrec-S, a model using only the extremely syntactic bit-level properties of Level-0; Dumrec-NT, a model using no Level-3 information (*i.e.*, no tips or points); and Dumrec, the full model.

Interestingly, Dumrec is reasonably good at gridfont letter recognition, most of the time either guessing the correct letter or picking a “reasonable” wrong one. Results of Dumrec’s performance are discussed below.

4.2 Netrec

We have also completed several experiments in gridfont letter recognition using very simple two- and three-layer feedforward connectionist networks. These “Netrec” networks are trained (using backpropagation) on a variety of gridletters and then tested on letters they have not yet seen. Results show that connectionist networks can perform fairly well on recognition tasks if letters are very normal (McGraw and Drasin, 1993). The ability to generalize about input patterns and to create complex association maps between input patterns and their categories is of course the key to gridletter recognition — in fact, to recognition of anything at all. However, due to the statistical, syntactic nature of their associations, networks trained on normal letters have a very hard time categorizing non-normal letters.

Our networks use a localist representation for the input of a gridletter. Although the representation is sparse, and provides no spatial-relationship information to the network, it is exactly the

consists simply of the number of quanta in each grid sector. A letter’s “gravity” is the location of the sector with the most quanta. A letter’s “mass-vector” is slightly more complicated. It consists of a vector with its origin in the lower left corner of the grid that extends into the grid according to the position and number of quanta making up the gridletter. This vector changes according to how much “stuff” there is in a letter, and where that “stuff” is located. Comparing these properties between gridletters requires some knowledge of the structure of the grid and the relation between sectors. For this reason, level-2 data are slightly more abstract than level-1 data.

Level-3 data constitute the most abstract data in the Dumrec program. Properties at this level include “point” and “tip” information. “Points” are the 21 intersection points of quanta on the grid. All the points where a gridletter terminates are recorded for the point property. For instance, most versions of the letter ‘c’ on the grid have two points associated with them, the locations where the ends of the (conceptual) curve are located. “Tips” are related to points, but also include information about the direction of termination and its type. Each tip is a quantum with a direction of termination and a connectedness to the rest of the letter. For example, the vertical ending of the part filling the *post* role on the left side of a very standard letter ‘b’ would be a vertical tip, pointed upwards, and connected to the rest of the letter at its bottom. Most instances of the letter ‘k’ have 4 tips, two vertical ones ending in different directions (either end of the part filling the *post* role), and two kinds of diagonal ones on its right side. The number, location, and type of tips in the mystery letter are compared with those of the prototype letters. This calculation involves a small amount of knowledge about gridletter orientation and the relationships between *types* of tips. Also note that it cannot be computed completely locally, and instead requires knowledge of the entire shape. Thus, level-3 data are the most abstract properties found in Dumrec (even if they are not very abstract), and require the most computation. Obviously, though, even the most abstract of the Dumrec levels is still far lower, conceptually speaking, than the level of roles.

Once the property data have been calculated for the mystery letter, Dumrec calculates a score for each training letter based on its relation to the mystery letter. The score is a weighted ranking of the match between property lists. Comparisons of properties range from the trivial (for level-0 data) to rather complicated (for level-3 data), but are all relatively simple computations. Obviously, the weights play a crucial role in Dumrec’s performance and are very important. The weights used for this experiment were set by hand over a very large number of runs (during which **TRAIN** was used as the prototype set). The same set of weights was used in each of the different Dumrec models reported below. Because Dumrec’s weights interact with each other in nonlinear and hard-to-predict ways, they cannot be tuned by trivial parameter-tuning algorithms. As has been discovered over and over again in machine learning, an intelligent user is often the best way to tune complex models. After raw scores are calculated, any of several different selection metrics (explained below) can be applied.

4.1.1 The three faces of Dumrec

This paper presents the results of three different Dumrec models. The models differ according to the properties utilized in computing a letter’s score (*i.e.*, level of the data allowed in processing). Table 5 shows all of the models arranged by type and introduces the names we will use to refer to them below.

Dumrec was originally intended as a winner-take-all (WTA) model in which the scores yielded a simple rank ordering of the prototypes. The selection rules we tested reflected this philosophy simply and easily. Under the WTA-*token* selection scheme, the prototype token (from **TRAIN**) with the highest score (meaning that its properties are in some sense the most closely matched to the mystery letter’s) wins, and the mystery letter’s category is presumed to be the same as that

The information gleaned from this process can then be used in the development of more sophisticated cognitive models.

4.1 Dumrec

As its name is intended to suggest, Dumrec is a brute-force symbolic-AI gridfont recognition program. It is written in Scheme (a dialect of Lisp). Dumrec compares a mystery letter with a large number of stored “prototype” letters, counting agreements and disagreements (which are computed at various levels of complexity), and making a category decision based on a weighted comparison of low-level features. Dumrec’s database of stored letters can be changed to include any number of different “training letters”. All of the Dumrec models reported here used the letters of the **TRAIN** data set as example letters.

Dumrec is initialized by reading in a data file containing a large number of training letters (several for each of the 26 letter categories) and computing a property list for each letter in the list. After this, Dumrec is given any number of mystery letters, one at a time. When a mystery letter is presented, its property list is computed in exactly the same way as for the training letters. The mystery letter’s property list is then compared, using a weighted metric, to the property lists of all of the training letters, and a winner is picked.

Dumrec uses two types of comparison. The first is a direct approach in which the quanta of the mystery letter are directly compared to the quanta of each training letter with no abstraction whatsoever. Dumrec has six comparisons of this type, involving what we call Level-0 data: a rating based on Hamming distance between the quanta lists of the mystery letter and the prototype letter, an “on”-bit rate, which is the number of quanta shared between the mystery letter and each of Dumrec’s prototype letters, and four simple scores based on the number of horizontal, vertical, forward-slash or back-slash quanta in a letter. Points are awarded to each training letter based on how well its properties match those of the mystery letter.

All other points awarded to training letters are based on comparisons of pre-computed feature lists of each training letter to the mystery letter’s feature list. These features can be separated into three levels (based on the amount of abstraction involved in their computation).

Level-1 data involve only a small amount of abstraction, as level-1 properties are concerned with sets of multiple quanta. Level-1 data include position and number information for a variety of simple angles and shapes. These data are computed by sliding simple shapes over the grid and noting matches. For example, a line segment composed of two horizontal quanta is slid down the grid. Each of the seven possible positions where such a line can occur on the grid gets a value of either 1 or 0 depending on whether the segment as a whole is present. The total number of such matches is also computed. The vector pertaining to horizontal-segment occurrence in a mystery letter can then be compared to the corresponding vectors already computed for the prototype letters, and a score assigned. Shapes that were considered include: $n \times m$ rectangles (where $n = [1..2]$ and $m = [1..4]$), all possible triangles made of sides with either 1 or 2 quanta, vertical line segments of length $[2..4]$, horizontal lines of length 2, forward and backward slashes of length 2, and all possible angles (*i.e.*, places where two quanta meet) formed on the grid of 45, 90, or 135 degrees. Also contributing to a prototype letter’s score are the total number of certain types of such shapes, including: simple closures, boxes, rectangles, triangles, lines, and angles. Such a brute-force approach to searching for shapes is easily implemented, given the relatively small number of quanta on the grid. Level-1 data are all entirely syntactic and locally calculable, but are slightly more abstract than level-0 data.

Level-2 data consist of slightly more abstract properties — namely, “mass”, “gravity” and sectorization data. The grid has six sectors (the three zones shown in Figure 5, divided into left and right halves), each of which may or may not include quanta from the letterform. Sectorization data

Results show that role-parsings are much preferred over Palmer-parsings in both shape and letter conditions (77.9% of the time in both conditions combined). It is possibly the case that with another set of regression weights, the goodness ratings of the role-parsing parts would be seen to be higher. We have not investigated this possibility.

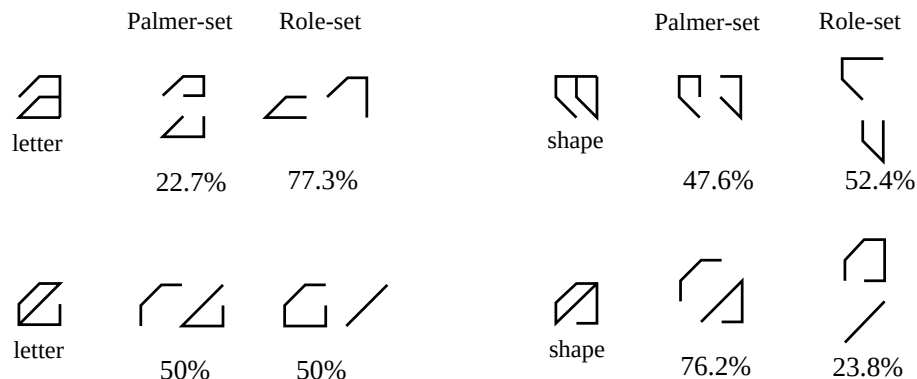


Figure 14: Two stimulus pairs from the part–role parsing experiment (letter condition on the left, shape condition on the right). The numbers shown below the stimuli are the percentage of subjects who chose the corresponding part-set as best. Note that the “same” parsings (rotated and/or flipped) result in different preferences, depending on the condition.

One result is especially relevant to the role hypothesis. Of the seventeen stimuli, there were two cases where the preference for role choices interacted strongly with the primary letter/shape condition. Figure 14 shows them both. In the case of the ‘a’, the role-parsing was preferred to a much greater extent in the letter condition than it was in the shape condition. In the case of the ‘e’, the Palmer parsing was preferred to a much greater extent in the shape condition than it was in the letter condition. Some formal models of segmentation, such as Palmer’s, suggest that the goodness ratings of parts should be the same in both conditions (since one is a simple rotation of the other). If this were the case, then the two parsings in either stimulus set should be picked the same number of times in both conditions. Instead there is a quite strong effect according to the condition. These two findings serve as further evidence of the top-down effect of roles on letterform perception.

4 Simple computer models of letter perception

The next two sections are concerned with computer models of letter perception. These models have been developed for two reasons. First, they are intended to serve as benchmarks against which the strengths and weaknesses of Letter Spirit’s recognizer can eventually be compared. There is little doubt that inter-architecture comparison and analysis of this sort will provide useful data for cognitive modelers.

Second, and most importantly for this paper, the rival models serve to shed some light on the human data collected (and vice versa). We suggest that comparing the performance of the simple models (which we can selectively lesion) with the performance of human subjects should be considered an interesting data-analysis technique in its own right. It is especially useful when data are as complicated and processing is as varied as is the case in letter perception. By selectively lesioning the models we can determine what factors influence people in letter-categorization tasks.

they will fit some letter-conceptualizations better than others). The problem is that too many sets of roles can be fit into poorly, so that role-filling is not much help, and finally, after an extended period of time, an informed guess of some sort is made. The two tokens in the “14-guons” column of Figure 13 are very strange tokens indeed, each having elicited fourteen possible categorizations.

The ambiguous/strange distinction is also evident in the fonts *sabretooth* and *close*. *Sabretooth* is a somewhat stranger font than *close*. In other words, *sabretooth*’s gridletters have parts that do not fill roles in a normal fashion, so the font as a whole has a higher average response time than *close* does, even though the accuracy rates of the two fonts are very similar.

3.6 Double tokens and the question of determinism

While most models of letter recognition are deterministic and take into consideration only the stimulus and the ability of the subject, our experiments show that human letter recognition is not deterministic, in the sense that presenting the same stimulus twice to a subject does not necessarily evoke the same response. Of the 545 tokens, there were eight pairs of tokens that happened to be identical. These tokens were presented twice to each subject, although the trials were of course separated randomly. The reaction times of the first exposures and the second exposures correlated surprisingly poorly ($r = 0.235, p > 0.01$). Not only the reaction times but also the category responses differed over the two exposures. The average agreement in responses between two trials from the same subject on the same (double) token was 90.0% — virtually the same as the overall accuracy of categorization (89.4%) on the entire set of double tokens. That is, the chance of a subject identifying any particular one of these tokens correctly *at all* is the same as the chance that both categorizations of that token will be the same. There does appear to be a facilitating effect in repeating stimuli, as the average response time for second exposures (1.24s) is significantly lower than average response time of first exposures (1.50s).

While this finding does not indicate that a successful letter-recognition system must be stochastic, it does show that humans achieve very good performance while being heavily influenced by factors beyond those that are commonly assumed to completely determine the outcome of a trial.

3.7 A follow-up experiment

A small experiment concerning the possible top-down effects roles have on the parsing of letterforms into parts was also carried out. Forty-three subjects were divided into two conditions — shape and letter. Letter condition subjects received stimuli that were always gridletters. Shape condition subjects received the same stimuli, except randomly rotated (through 90 or 180 degrees) and/or flipped. Each group was asked to choose between two alternative parsings of a stimulus. Of the two possible parsings, one was deliberately created with parts that filled the letter category’s roles extremely well, while the other was deliberately created not to match roles at all. However, the poor role-filling parsings were made in such a way that the goodness ratings of their parts were much higher than the goodness ratings of the role-parsings’ parts. Goodness ratings for parts were computed using Palmer’s part-rating model (previously mentioned in Section 1.1) with the regression weights reported in (Palmer, 1977).¹¹ A goodness rating in Palmer’s model is computed as a combination of properties of proximity, connectedness, orientation, length, and continuity — all gestalt features with no top-down influence.

¹¹One important caveat: Palmer claims that the regression weights need to be recomputed for each particular set of stimuli. We did not do this. Instead we used the reported weights. We feel that the plasticity of the weights seriously weakens Palmer’s model since reweighting the six high-level properties will almost always result in a fit to experimental data.

non-standard letterforms and is probably magnified under time pressure. In the case of ‘l’ seen as ‘i’, changing the default height of ‘l’ to fit the unusually short token is thrown out in favor of filling the dot role of ‘i’ by the top part of the higher-than-usual token (or perhaps perceiving the token as an uppercase ‘I’).

The second class of error includes ones in which subjects either made reference to categories outside the stated limits of the experiment (*e.g.*, capital letters) or violated the placement constraints of the grid. Both are types of error that the computer models discussed below cannot make. An example of this is making a classification based on a resemblance to an uppercase letter (as in ‘d’ seen as ‘J’, above) or overriding the conventions of the grid itself (as in ‘e’ seen as a very tiny ‘p’ with no quanta in the “official” descender zone). We consider the “official” ascender and descender zones as the only proper place for ascenders and descenders, but not all of our subjects were like-minded.

3.5 Prediction 3

Every token elicited a certain number of responses from the group of subjects. While some tokens were always identified correctly (thus placed in only one category), others were occasionally misclassified but with only one wrong answer given instead of the correct answer (for a total of two categories). Still other tokens were categorized in several different ways. As would be expected, there is a strong correlation between the number of categories in which each token was found and its average reaction time ($r = 0.85, p < 0.01$). Fast response times are seen when there are few plausible category bins. As the number of plausible answers increases, response time becomes longer.

The number of categories implies a continuum that we might call “n-guity”. A token that is always identified correctly has parts with a high “goodness-of-fit” to the roles of only one category. A token whose responses are divided between two categories could be considered ambiguous. As a token becomes stranger, the obviousness of any one or two answers will be lost in the crowd of poor part–role fillings, and the number of responses will grow quite large — as high as fourteen in our experiment.

ambiguous	3-guous	14-guous
 i 1.43	 a b d 2.70	 a b c e f G n o Q r s v x z 6.80
 q g 1.60	 z t x 2.33	 b c e i j m o s t u v w y z 7.38
 v u 1.57		

Figure 13: Ambiguous and strange letterforms (14-guous being the strangest of all). Beside each gridletter are the categories people guessed and the average response time.

Ambiguity and strangeness both cohere nicely with the role hypothesis as anticipated in prediction 3. When a token is ambiguous, its parts can fill two or more sets of roles all in a fairly reasonable fashion. This is illustrated by the first two columns of gridletters in Figure 13. A choice must be made between the possibilities. This choice takes longer as more competing categories are considered. When a token is strange, its parts do not really fit any sets of roles well (although clearly

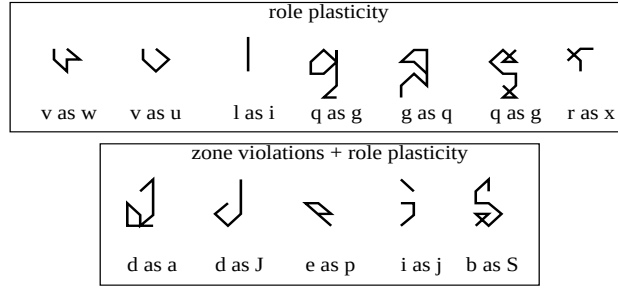


Figure 12: Examples of some commonly miscategorized tokens. These can be divided into two classes as shown. Note how reasonable these errors are.

people pay special attention to such closure or its absence when they are categorizing ‘b’s and ‘h’s. Similar stories can be told about ‘c’ and ‘o’, about ‘m’ and ‘n’, and about ‘e’ and ‘s’. These pairs of letters are rarely confused by people because the conceptual differences are too great.

Next, consider the most common human errors not found in the list of most syntactically-similar pairs. The errors include: $u \leftrightarrow v$, $i \leftrightarrow l$, $i \leftrightarrow j$, $f \leftrightarrow t$, $v \leftrightarrow w$, $c \leftrightarrow r$, and $j \leftrightarrow s$. Consider the default roles for ‘v’ and ‘u’. As the “point” of a ‘v’ (at the bottom) becomes rounded, the letter becomes more ‘u’-like. Similar role-level analyses hold for other letters: a ‘j’ can become more ‘i’-like as its descender gets straighter and shorter; an ‘f’ can become a ‘t’ if the stem is bent differently; an ‘i’ can become more ‘l’-like as the space between its dot and its stem gets smaller; a ‘v’ becomes a ‘w’ if its point gets bent upwards instead of down. These are several examples of role plasticity that are in some respects brought to the forefront by the constraints of the grid. Similar stories can be told for the remaining common errors not found in the syntactic clustering. Because roles are plastic, human letter perception is flexible and extremely reliable. The plasticity comes, however, at a price of occasional error-making. By closely examining the errors, we can come to understand the relationships between letter-concepts.

We argue that *all* of the most common human category errors shown in Table 4 can be understood in terms of role plasticity. Precisely because of the plasticity of roles, ambiguous or strange role-fillers can be accommodated and seen as instances of roles. Syntactic proximity is clearly not as powerful as an explanatory method (as anticipated in prediction 2). That errors are easily explained by the role hypothesis becomes even more obvious when specific errors, rather than category-level error rates, are considered. At the token level, all common human errors can be seen to be role-level errors. We look at this effect next.

Figure 12 shows a number of commonly misperceived tokens from **ALL**. The first thing to note is that all of the errors are fairly reasonable — in fact, it is not clear that some of them are “errors” at all.¹⁰ People make an occasional dumb mistake, but most category misidentifications can be easily explained in terms of conceptual proximity. Generally, there are two types of problems causing miscategorization. The first type involves perceptual mistakes where non-standard parts that serve as role fillers cause miscategorization. In the example shown, the roles of a ‘w’ are filled quite nicely by what was supposed to be a ‘v’ — with the kink on the right side being seen as forming the bottom hump of a ‘w’. Such errors are commonly caused by the coarseness of the grid, which often forces letterforms to the fringes of category space.

In some cases, the perceiver allows very poor parts to be seen as role-fillers in order to make sense of weird shapes, resulting in miscategorization. This usually occurs with difficult or otherwise

¹⁰See footnote 7.

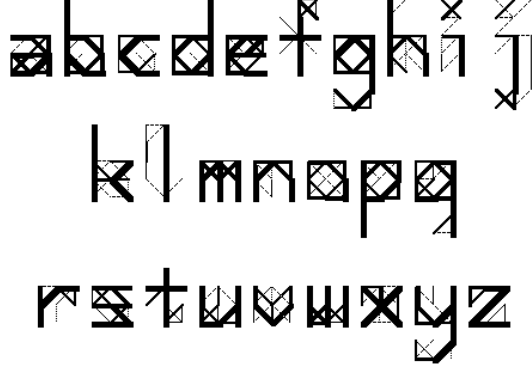


Figure 11: The blurred prototypes of **ALL**. The thickness of a quantum indicates its relative frequency.

syntax	
Error	Error Rate
$m \leftrightarrow n$	8.80
$n \leftrightarrow r$	8.35
$u \leftrightarrow w$	7.15
$c \leftrightarrow o$	5.75
$b \leftrightarrow h$	4.90
$g \leftrightarrow q$	4.80
$e \leftrightarrow s$	4.60
$e \leftrightarrow o$	4.55
$n \leftrightarrow o$	4.30
$h \leftrightarrow n$	4.15
$a \leftrightarrow o$	4.15

Table 3: The eleven closest categories as calculated by the city-block distances between the blurred prototypes of **ALL**. The bidirectional error rate is taken from the associated prediction matrix (see Section 6). These pairs give some idea of the syntactic relationship between categories of **ALL**.

humans	
Error	Error Rate
$u \leftrightarrow v$	16.30
$g \leftrightarrow q$	10.95
$i \leftrightarrow l$	7.95
$i \leftrightarrow j$	7.45
$f \leftrightarrow t$	5.75
$v \leftrightarrow w$	4.75
$h \leftrightarrow k$	4.50
$n \leftrightarrow r$	4.05
$c \leftrightarrow r$	3.30
$j \leftrightarrow s$	3.10

Table 4: The top 10 category errors made by humans. The bidirectional error percentages are computed from the confusion matrix shown in Table 1.

3.4 Prediction 2

Table 3 displays the eleven category pairs that are syntactically closest (by overlap of quanta) in the data set **ALL**. The list is meant to provide some insight into the low-level syntactic similarities between letter categories in **ALL**. It was created as follows. The *blurred prototypes* of **ALL**, graphically represented in Figure 11, were first created by averaging the binary quanta lists of all of the letters in a given category together, thus providing a value between 0 and 1 for each of the 56 quanta. This resulted in a 56-dimensional vector for each category. Given any two such vectors, the city-block distance between them (*i.e.*, the sum of the absolute values of their 56 coordinate-pair differences) gives a low-level syntactic measure of category similarity.

Despite the apparent naivete of the blurred-prototype model, the approach works to some degree. This is evidenced by the performance of a recognition program that identified tokens as belonging to the category that their syntax matched best. This simple model (essentially a variation on the Dumrec-S program we describe later) recognized the tokens in our experiment at a reasonable rate (55.9% recognition on ALLbutTRAIN). Similar models, deviating from the single-mindedness of this approach only by allowing 2-D translation of images or pixels, form the basis of some OCR models in the literature (Mantas, 1986; Mori et al., 1984; Hinton et al., 1992; Gaillat and Berthod, 1979).

It is interesting to contrast the syntactically determined category-pairs in Table 3 with the most common human category errors, shown in Table 4. Although some human errors seem to be due to simple syntactic proximity of letter shapes, many are not. The human errors that can be explained or predicted in terms of syntactic proximity are $n \leftrightarrow r$ and $q \leftrightarrow g$. Note, however, that both of these letter pairs share many common roles as well. For instance, ‘q’ and ‘g’ both have a bowl or a circle in the central zone and have a descender on the right. Similarly, ‘r’ and ‘n’ both have a stem on the left that goes from the baseline to the x-height, as well as a role that attaches to the stem near its top on the left and arches over to the right. It is this role-level similarity of ‘r’ and ‘n’ that can cause ambiguity. Depending on how close the filler of the latter role gets to the baseline, the amount of ‘n’-ness of the letterform will vary. Such structural similarities are suggested by Figure 2. In fact, we argue that it is the role similarities that give rise to the shape proximities in these cases, and that confusion in these cases is a result of higher-level role similarities (as stated in prediction 2).

The cases where syntactic proximity differs from human errors are more interesting. First, consider the pairs in Table 3 that do not correlate with high-frequency errors in human perception. The following pairs have great proximity in syntactic space but occur only rarely as errors (the numbers in parentheses): $m \leftrightarrow n$ (2.7%), $u \leftrightarrow w$ (0.9%), $c \leftrightarrow o$ (0.7%), $b \leftrightarrow h$ (1.3%), $e \leftrightarrow s$ (0.2%), $e \leftrightarrow o$ (0.4%), $n \leftrightarrow o$ (0.5%), $h \leftrightarrow n$ (2.3%), and $a \leftrightarrow o$ (2.3%). People hardly ever make these errors even though the letters are syntactically similar at the quanta level. This result supports prediction 2.

Our results corroborate the work of Medin et al. (1993) and Goldstone et al. (1991), who show that literal physical properties are often less important in similarity judgments than are higher-level relations. This particular result is also compatible with all accounts of categorization that consider emergent features (*e.g.*, closure) (Treisman and Paterson, 1984). However, since the role hypothesis is concerned with structured representations made up of emergent features — as each role is comprised of norms governing groups of emergent features, and r-roles provide information about the interaction between roles — it goes a step beyond standard emergent-feature models such as Treisman and Paterson’s.

We hypothesize that for any pair of syntactically similar letters, there are salient differences in their role-sets that cause people not to confuse them. For example, the difference between ‘b’ and ‘h’ is simply a matter of closure at the bottom, but this seemingly small difference plays a very significant role in people’s categorization processing, precisely because they know of the existence (and the conceptual proximity) of the other category. That is, the closeness of the role sets makes

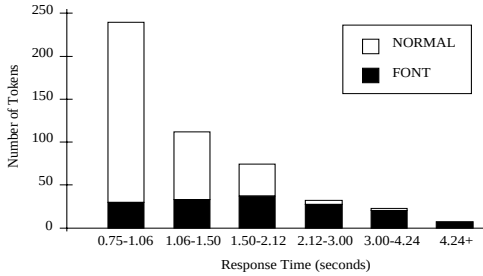


Figure 8: Each column shows the total number of tokens that occurred within certain response time ranges. Tokens are separated into **FONT** and **NORMAL** groups that (stacked together) sum to the column total.

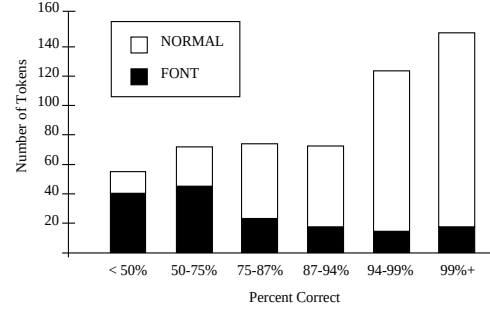


Figure 9: The number of tokens, separated into **NORMALS** and **FONTs** groups (as in the previous figure), that occurred within certain accuracy ranges.

Table 2. As a result, response time does not correlate highly with the number of quanta ($r = 0.097, p < 0.1$); nor does accuracy ($r = 0.18, p < 0.01$).⁹ The **NORMALS** and **FONTs** data of Table 2 make this especially obvious — note that both *close* and *hint four*, which have far fewer quanta on average than the **NORMALS**, are clearly harder to recognize. The role hypothesis can account for this result as follows: some gridletters with a higher-than-average quanta count still have well-filled roles, while others having an average number of quanta, or possibly even a lower-than-average quanta count, have troublesome role-fillers. This effect is also apparent within the **FONTs** group, with *sabretooth* having fairly strong but heavy role-fillers (making it relatively easy to recognize) and *double backslash* having roles filled in less standard ways (making it harder to recognize). Of course, most **NORMALS** tokens, having very normal role fillers, are even easier to recognize than the *sabretooth* tokens. Figure 10 shows some specific tokens (as well as accuracy and response-time data) from *sabretooth*, *double backslash*, and **NORMALS** to illustrate this point.

40%	50%	100%	14.3%	64.3%	94.3%	37.1%	71.4%	100%
1.675s	2.125s	0.94s	2.856s	2.751s	1.140s	7.376s	3.686s	1.071s
74.3%	100%	94.3%	22.9%	100%	85.7%	45.7%	78.6%	97.1%
3.720s	1.063s	1.154s	2.512s	1.585s	2.037s	1.999s	2.479s	1.135s

Figure 10: Standard vs. non-standard role-fillers as exhibited in *double backslash* (left letterform of each triplet), *sabretooth* (middle), and tokens from **NORMALS** (right). There are six sets of triplets, one for each of the letters ‘e’, ‘f’, ‘i’, ‘k’, ‘l’, and ‘u’. Starting at the left of each triplet, where roles are poorly filled, roles are filled in a more standard fashion towards the right. Role-related difficulty is reflected in the percent-correct and response-time data.

⁹ Again, these correlations are made over all tokens.

Stimulus set	accuracy	response time	# of quanta
ALL	84.0	1.84	6.80
NORMALS	90.1	1.46	6.97
FONTS	68.7	2.80	6.36
<i>bowtie</i>	52.1	3.79	7.65
<i>double backslash</i>	66.2	2.88	6.00
<i>flournoy ranch</i>	57.4	3.44	7.54
<i>hint four</i>	85.3	1.95	4.92
<i>close</i>	74.8	2.12	5.31
<i>sabretooth</i>	76.7	2.61	6.73

Table 2: Human recognition performance averaged over 35 subjects.

been shown to shed light on letter similarity (Gilmore et al., 1979), it is informative to note the most common perceptual errors and to try to understand just why they occur. Many cognitive scientists have noted the importance of studying error-making in order to shed light on cognitive processing (Norman, 1988; Dell, 1986; Cutler, 1982; Fromkin, 1980).

3.3 Prediction 1

Table 2 shows average percent correct and response-time values for some of the data subsets. For all groups, with the exception of the gridfonts *sabretooth* and *close*, accuracy and speed have corresponding ranks. In general, correct responses came more quickly than errors, but the letters that were identified the most quickly were not necessarily those identified correctly most often. Generally, as response time increases, accuracy decreases ($r = 0.74$, $p < 0.01$ for a correlation over all tokens averaged over all subjects). Both of these measures can be understood in terms of a third variable corresponding to an intuitive notion of *difficulty*, with response time rising and accuracy falling as difficulty rises.

Difficulty can be explained in terms of the ease with which a letterform can be parsed into parts that fill roles. Complicated letterforms take longer to parse into roles, causing response time to rise. Likewise, letterforms with parts that match sets of roles poorly or ambiguously cause more errors than letterforms with parts that easily fit within the norms provided by sets of roles. Figure 3 provides an example of “good” role fillers vs. “bad” role fillers. The role hypothesis is thus consistent with the relationship between response time and accuracy, as stated in prediction 1.

Of all the gridfonts, the one with the hardest letters to recognize was the complicated *bowtie*, in which the style pushes the gridletters to the fringes of category space. The easiest complete font was the relatively simple *hint four*. Easier than the full fonts were the letters belonging to the class of **NORMALS**, which, of course, we picked for their normalcy (see Figure 19). There is a striking difference in response time (and accuracy) between **FONTS** and **NORMALS**. Figure 8 shows a histogram of the number of tokens in the data set that fell into certain response-time bins. It shows that most of the tokens in **NORMALS** are recognized quickly while most of the **FONT** tokens take longer. Figure 9 shows a similar histogram with accuracy along the x-axis. Clearly, most **NORMALS** tokens are recognized more accurately than the **FONT** tokens are. These two effects also hold within the **FONTS** category itself, with some fonts having a lower recognition rate while at the same time taking longer to recognize. The **NORMALS** generally have standard parts that easily fill roles whereas **FONTS** have non-standard role-fillers. The fact that **NORMALS** are more easily recognized than **FONTS** lends further support to prediction 1.

The number of quanta in a gridletter does not alone determine its difficulty, as is apparent in

stimulus	response																									
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	.833	.005	.000	.026	.019	.000	.011	.000	.000	.005	.000	.000	.000	.000	.018	.000	.011	.003	.002	.002	.010	.011	.018	.011	.000	.013
b	.002	.910	.000	.005	.002	.000	.002	.005	.000	.002	.005	.009	.000	.000	.000	.003	.000	.000	.046	.000	.002	.005	.000	.002	.000	.002
c	.014	.000	.884	.002	.012	.000	.025	.000	.002	.000	.000	.006	.000	.000	.010	.000	.000	.002	.002	.000	.012	.017	.008	.004	.000	.002
d	.035	.011	.000	.884	.000	.000	.013	.000	.000	.030	.002	.000	.000	.000	.000	.000	.003	.000	.008	.000	.003	.003	.002	.002	.000	.003
e	.005	.000	.012	.001	.912	.000	.010	.000	.000	.000	.000	.000	.000	.000	.000	.033	.014	.000	.001	.000	.000	.000	.000	.000	.001	.010
f	.002	.000	.000	.002	.034	.732	.005	.025	.009	.018	.002	.004	.022	.000	.000	.007	.009	.005	.007	.108	.000	.002	.004	.000	.002	.000
g	.003	.000	.000	.000	.000	.003	.906	.003	.000	.009	.000	.000	.000	.000	.000	.003	.040	.000	.025	.000	.000	.000	.000	.000	.004	.000
h	.000	.021	.004	.000	.006	.002	.008	.896	.000	.000	.008	.000	.002	.023	.000	.000	.000	.000	.010	.002	.002	.002	.000	.002	.010	.000
i	.000	.004	.004	.000	.003	.019	.013	.000	.745	.102	.001	.001	.003	.001	.006	.004	.004	.006	.046	.017	.001	.003	.004	.004	.003	.001
j	.001	.004	.000	.001	.001	.001	.008	.001	.047	.862	.001	.002	.000	.000	.000	.000	.001	.000	.061	.002	.001	.001	.000	.001	.003	.000
k	.000	.011	.005	.000	.005	.029	.006	.082	.000	.000	.786	.006	.003	.003	.000	.000	.000	.002	.008	.003	.003	.006	.002	.037	.002	.000
l	.000	.029	.041	.000	.002	.015	.004	.004	.158	.021	.004	.593	.000	.000	.000	.000	.004	.008	.031	.033	.025	.004	.000	.002	.000	.023
m	.000	.000	.000	.000	.008	.004	.000	.001	.000	.001	.001	.001	.928	.012	.000	.001	.000	.000	.001	.001	.001	.000	.001	.019	.018	.000
n	.029	.004	.002	.005	.004	.002	.004	.022	.004	.004	.000	.000	.042	.817	.005	.000	.000	.011	.002	.013	.000	.000	.002	.016	.005	.007
o	.027	.015	.003	.053	.008	.000	.000	.000	.003	.005	.002	.000	.000	.000	.828	.002	.012	.003	.003	.000	.011	.020	.000	.002	.000	.000
p	.000	.000	.000	.000	.013	.000	.017	.000	.000	.000	.000	.001	.000	.000	.001	.936	.020	.007	.000	.000	.000	.000	.000	.000	.004	.002
q	.003	.002	.001	.001	.001	.000	.179	.000	.001	.002	.000	.000	.000	.000	.002	.013	.775	.000	.006	.000	.000	.000	.001	.000	.001	.002
r	.012	.005	.064	.000	.011	.003	.002	.006	.009	.003	.003	.000	.000	.070	.005	.020	.002	.701	.009	.012	.000	.005	.000	.030	.026	.005
s	.004	.000	.000	.000	.003	.000	.004	.000	.001	.001	.000	.001	.000	.005	.003	.000	.001	.004	.935	.000	.004	.010	.001	.004	.005	.011
t	.003	.010	.024	.000	.005	.007	.010	.000	.007	.002	.007	.007	.000	.002	.002	.000	.003	.012	.000	.866	.003	.000	.002	.015	.010	.003
u	.009	.003	.004	.001	.003	.001	.003	.001	.000	.003	.006	.001	.000	.001	.003	.000	.000	.004	.001	.003	.792	.127	.014	.003	.014	.000
v	.000	.009	.017	.001	.019	.000	.001	.000	.000	.001	.000	.000	.000	.000	.000	.009	.000	.000	.003	.001	.199	.654	.084	.001	.000	.000
w	.000	.000	.000	.000	.010	.000	.000	.000	.000	.000	.010	.001	.001	.000	.000	.000	.002	.000	.001	.000	.003	.011	.937	.021	.002	.000
x	.002	.000	.000	.000	.000	.003	.000	.002	.000	.002	.006	.000	.010	.002	.000	.000	.000	.003	.008	.021	.000	.000	.000	.929	.003	.008
y	.001	.000	.000	.000	.003	.000	.004	.000	.000	.002	.000	.000	.000	.000	.000	.000	.002	.000	.004	.001	.007	.001	.004	.001	.965	.003
z	.004	.000	.023	.004	.023	.000	.008	.002	.004	.000	.000	.000	.000	.000	.002	.000	.008	.006	.014	.017	.010	.008	.004	.039	.031	.795

Table 1: Confusion matrix of human performance on ALL.

NORMALS: This is the union of **TRAIN** and **NON-TRAIN** (and is shown in Figure 19).

FONTs: This consists of six entire gridfonts (shown in Figure 20), each a coherent set of 26 tokens meant to share the same stylistic identity (or *spirit*). The styles of these fonts ranged from eccentric and highly stylized to reasonably normal.

ALLbutTRAIN: The **ALL** data set minus the **TRAIN** data set (put another way, the **FONTs** dataset plus the **NON-TRAIN** dataset.) This subset was used in program/human comparisons.

Subjects were told that they would see a series of stimuli, each of which was intended to represent a lowercase roman letter. Responses were to be rapid keyboard keypresses that identified the stimulus as one of the 26 lowercase roman letters; identification of a stimulus as any other character in any other alphabet was disallowed. The importance of correct identification was stressed over that of speeded response, although subjects were aware that their response times were being recorded. It should be noted that keyboards are labeled with uppercase letters, and so do not provide novel prototypes of lowercase letters.

For 21 of the subjects, the data set was **NORMALS** plus four of the gridfonts from **FONTs** (104 more tokens, for a total of 389): *bowtie*, *double backslash*, *flournoy ranch*, and *hint four*. Two additional gridfonts (*close* and *sabretooth*) were added for the final 14 subjects only (for a total of 441 tokens). The order of presentation was uniquely randomized for each subject, to alleviate possible context effects. Subjects were not made aware that any letters were drawn from stylistically consistent gridfonts (in **FONTs**).

Subjects. 35 subjects were drawn from the Indiana University community and included undergraduates, graduate students, and staff. None were involved in work related to gridfonts, or fonts of any kind. All subjects were compensated for their participation and were debriefed after the experiment.

3.2 General results

Our experiment involved the presentation of 17,973 trials divided over the 35 subjects. 27 of the trials were thrown out — one due to an impossibly short response time (which occurred due to a keyboard entry error) and 26 because of illegal responses (*e.g.*, typing ‘9’ instead of a valid letter). Subjects varied between 63.7% and 95.6% correct responses with a mean of 84.0% correct.⁷ Average response times ranged between 0.84 seconds and 3.12 seconds with a mean of 1.84 seconds. In individual trials, response times varied between 0.47 seconds and 56.4 seconds.

Table 1 shows a stimulus-by-response confusion matrix generated for the data set **ALL** by averaging over the 35 subjects and collapsing tokens into their respective categories. The table shows the percentage of the time that tokens belonging to a particular category were categorized in each of the 26 response categories. Correct-response percentages can be read down the diagonal. The unidirectional error rates shown in the table can be averaged together to provide a bidirectional category error rate. As an example, consider ‘a’s and ‘b’s. ‘a’ tokens were properly categorized 83.3% of the time, and ‘b’s 91.0% of the time. ‘a’s were taken for ‘b’s 0.5% of the time while ‘b’s were taken for ‘a’s 0.2% of the time. The resulting bidirectional category error $a \leftrightarrow b$ is thus 0.35%.

Our confusion matrix differs considerably from others like it in the psychology literature in two ways. First, it reports on lowercase roman letters instead of uppercase.⁸ Second, it includes data about a variety of alphabetic styles instead of just one. Even though our data set is made up of arguably idiosyncratic gridletters, our stimuli are more realistic than many past data sets, in the sense that they capture more of the natural variability found among letterforms. This leads to a more thorough treatment of errors than has been evident in past work. As confusion matrices have

⁷In order to count as a “correct” response, a subject must have responded with the *one* unique category in which a stimulus was placed by the experimenters (the category intended by the designer of the given gridletter). For some of the more ambiguous tokens (see especially Figure 12), categorizing turns out to be a very hard task. We use the designer’s intended category in order to provide a fair comparison between humans and the recognition programs discussed below. Use of the term “correct” below will always be subject to this disclaimer.

⁸Townsend (1971), among others, reports on an uppercase confusion matrix.

letters that populate the fringes of a letter category. According to our theory of roles, human letter perception should be affected in the following ways:

1. Reaction time should be longer and recognition rate lower for complicated stimuli, in which roles are either ambiguously or poorly filled, than for relatively normal letters, with regular role-fillers. In other words, letters with role-fillers on the edges of acceptability should both take longer to recognize and be recognized less successfully, on average, than letters that have normal role-fillers. For example, the top two letterforms of Figure 3 should be more quickly and successfully recognized than the lower two. That complicated letters are harder to recognize is not a surprising result. What is important for our purposes is that our experimental observations are compatible with the role theory.
2. Low-level syntactic proximity (*i.e.*, degree of overlap of the quanta of a test letterform and the quanta of prototypes) alone should not account for errors in human letter recognition. Instead, errors should be explainable in terms of role plasticity, parts that fill roles, and relations between letter-conceptualizations. Category errors made by human subjects should be most prevalent between letters with similar sets of roles — roles that are defined by similar sets of norms. The roles shown in Figure 2 suggest why ‘n’s and ‘r’s, or ‘q’s and ‘g’s, should be easily confused whereas ‘n’s and ‘m’s, or even ‘e’s and ‘s’s (which are very often syntactically similar but are not closely related at the role level) should not.
3. The timing of particular errors should reflect the underlying role structure of letters. For instance, errors made in classifying letterforms with parts that fit reasonably well into one or more sets of conceptual roles should be made more quickly than errors made in classifying letterforms with parts that are too eccentric to fit any role-set well. The more ambiguous the parts, and the more tenuously they fill roles, the more time-consuming and less accurate the classification should be.

3.1 The experiment

Method

Stimuli. A black grid was displayed against a white background on a Macintosh screen. The 21 vertices were drawn with a unit distance of 1.31 cm, making the entire grid 7.86×2.62 cm. A solid square with sides of 0.13 cm was drawn at each vertex. Every legal quantum in the grid was drawn in with a line thickness of 0.04 cm. Thus the grid provided noise for each gridfont letter (henceforth, *token*) displayed. A token was displayed by drawing its active quanta with a line thickness of 0.13 cm. One example from the set of stimuli is shown in Figure 7. Subjects were seated, but able to maintain whatever distance from the monitor they found best.

Procedure. A data set of gridfont characters was shown to each subject.⁶ The tokens were presented individually, with the screen blank for two seconds between each response and the next stimulus onset.

The full data set, **ALL**, may be meaningfully separated into several non-exclusive subsets:

TRAIN: 130 tokens, which represent each of the 26 roman lowercase letter categories exactly 5 times each. These were intended to be very central examples of each letter. Computer models of letter perception that were created and tested as a supplement to the human experiment were trained using this group. The first five tokens in each letter category shown in Figure 19 make up **TRAIN**.

NON-TRAIN: These 259 tokens represented each letter category from 4 to 19 times. These were also intended to be very strong letters. For human subjects, this set should be qualitatively very similar to **TRAIN**, and differs from **TRAIN** only from the perspective of training the programs.

⁶All gridletters used in this experiment were designed by Douglas Hofstadter, Jim Marshall, and Gary McGraw.

clear). Nevertheless, most (but not all) gridletters are easily recognized by humans. One important feature of the grid is that it can both force letters to their categorical extremes because of its sparseness and yet still contain many easily identifiable letterforms. This makes it an excellent vehicle for letter-perception research — one that allows attention to be focused on issues of style. The domain was designed as an efficient medium in which to probe the conceptual nature of letterforms, with the hope that this will yield insight into what imbues concepts *in general* with their fluidity.

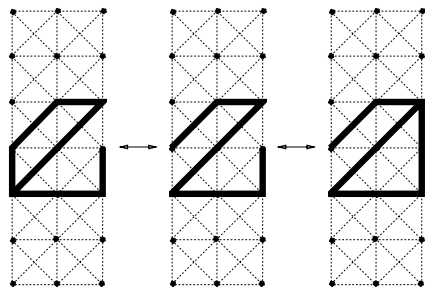


Figure 6: Metamorphosis of an ‘e’ into an ‘a’ through ‘z’ with a one-quantum change from one letter to the next.

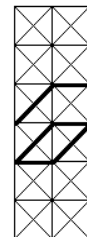


Figure 7: A sample stimulus (captured directly from a Macintosh screen). The presence of the grid provides some degree of noise. This gridletter is an ‘a’.

3 Results from human study

The experiment reported below is concerned with the recognition of gridletters. The purpose of the experiment is twofold: to gather data for comparison with cognitive models of letter perception, and to shed light on the human perception of lowercase roman letters that are stylistically eccentric. It is important to keep in mind the difference between the task we are asking people to do and other tasks that are related to letter recognition, like reading or unlimited-time categorization. Our task focuses on the quick recognition of single, isolated letterforms, without respect to word-level or other context effects.

Letters come in myriad styles, ranging from fairly bland (like the Times typeface in which this article is set) to highly stylized (like the ‘a’s of Figure 1). We are interested in the ways in which people perceive all kinds of letters — including ones that are eccentric. In designing our data set, we created two large groups on either end of the continuum of style. Each of the two groups — one with prototypical, non-eccentric, vanilla-style letters, and the other with non-standard, eccentric, style-filled letters — has many diverse letters as actual members. In the “normal” group there are some letters that are more daring than their compatriots, and in the “eccentric” group there are some letters that flirt with normality. The validity of our subjective division of these letterforms into the two groups is confirmed by the objective data that we gathered in the course of the experiment.

We now proceed to discuss the results of our experiment in terms of the role hypothesis. Although our results are compatible with the role hypothesis, they are also compatible with other models of letter perception. Stronger empirical evidence for the role hypothesis is presented in Section 6.

Our central predictions all relate to the hypothesis that letter concepts are comprised of simpler conceptual *roles*. The abstract, conceptual nature of roles is what allows people to easily recognize

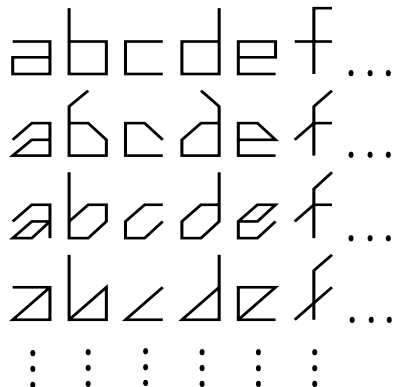


Figure 4: Items in any column have *letter* in common. Items in any row have *spirit* in common.

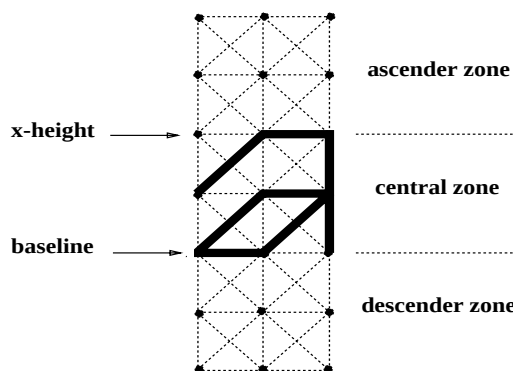


Figure 5: The Letter Spirit grid, with one of the possible sets of quanta instantiating an ‘a’ turned on.

perception and evaluation.

Implementation of Letter Spirit is in preliminary stages. Early work focuses on the recognition of gridletters. For more information about Letter Spirit, see (Hofstadter, 1987; McGraw, 1992; Hofstadter and McGraw, 1994).

2.2 The grid

In the Letter Spirit domain, letterforms are restricted to short line segments on a fixed grid of 21 points arranged in a 3×7 array (Hofstadter, 1985). Legal line segments, called *quanta*, connect a point to any of its nearest neighbors. There are 56 possible *quanta*, as shown in Figure 5. Because quanta are either on or off, decisions on the grid are coarse. Surprisingly, the variety among letters of a given category is still huge — hundreds of versions of each letter and several hundred full gridfonts have been designed by humans.⁵ Somewhat counterintuitively, the domain’s limitations engender this diversity.

It is impossible to make tiny changes on the grid, so any two instantiations of a given letter are significantly different. Indeed, the coarseness of the grid forces gridletter designers to tamper with the conceptual essence of letters. Such conceptual-level tampering makes the task of recognizing the final products that much harder. The addition or subtraction of even one quantum often fundamentally affects category membership. Figure 6 shows the large distance in letter-space one can travel with minimal changes on the grid. A “small” change — the erasure of just one quantum — changes the first shape from a strong ‘e’ to a strong ‘z’. A one-quantum addition transforms the ‘z’ into an ‘a’. So category membership in the Letter Spirit domain is a tricky matter. In fact, modeling the process of deciding which of the 26 categories a given shape on the grid belongs to (if any) is one of the hardest aspects of the project and is the focus of this article.

Because there are no fine-grained features to manipulate, gridfont designers often end up playing at the boundaries of the 26 categories. Consequently, many gridfonts are wild, sometimes having angular, blocky, spiky, sparse, or otherwise bizarre letters (as Figures 19 and 20 make abundantly

⁵The six human-designed gridfonts used in the experiment reported below, as well as a set of very standard gridletters, are displayed at the end of this paper. For more, see (Hofstadter, 1987).

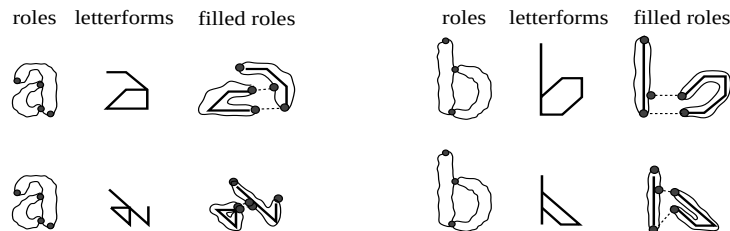


Figure 3: Parsing letterforms into high-level parts occurs under the top-down influence of roles. The top two of these four examples show straightforward parsings (or “good” *role-fillings*) involving little or no norm violation. The bottom two letters are more complicated, and have “bad” role-fillers. The roles must be plastic enough to accommodate the somewhat nonstandard parts of the bottom two letters. This plasticity comes from the fact that roles can tolerate a certain amount of norm violation on the part of their fillers.

1.5 Toward a conceptual approach

Our aim in this article is to show that roles play an important part in human letter perception.³ This becomes especially apparent when a letter is complicated or eccentric (*i.e.*, near the boundaries of its intended category). We are motivated by the need for a reasonable cognitive model of letter perception, which we intend to use as part of the Letter Spirit model of creativity in letter design. The Letter Spirit project is briefly introduced in Section 2. Letter Spirit will include a computer model of letter recognition based on the role hypothesis developed below.⁴ In Section 3, we discuss a psychology experiment in letter perception and relate the results to the role hypothesis. Section 4 introduces two different cognitive models of letter recognition — Dumrec, a prototype-based, brute-force-AI approach, and Netrec, a simple connectionist approach. In Section 5, the performance of the models is compared and contrasted with human performance. In Section 6, we introduce a set of theoretical prediction matrices of letter perception and use them to analyze the programs’ performance, as well as that of humans. Section 6 is critical, in that it provides our strongest evidence in favor of the role hypothesis. Section 7 discusses the role hypothesis in relation to other machine-based recognition approaches and psychological theories of recognition. We then go on to briefly sketch a computer model that we are currently implementing.

2 The gridfont domain

2.1 Letter Spirit in a nutshell

The specific focus of the Letter Spirit project is the creative act of artistic letter-design. The aim is to model how the 26 lowercase letters of the roman alphabet can be rendered in many different but internally coherent styles. Two important and orthogonal aspects of letterforms are basic to the project: the *categorical sameness* possessed by instances of a single letter in various styles (*e.g.*, the letter ‘a’ in Times and Optima) and the *stylistic sameness* possessed by instances of various letters in a single style (*e.g.*, the letters ‘a’ and ‘k’ in Times). Figure 4 shows the relationship of these two ideas. The program will start with one or more seed letters representing a style and create the rest in such a way that all 26 share the same style, or *spirit*. Letter Spirit will do all of its own letter

³A preliminary version of these experiments was presented in (McGraw et al., 1994).

⁴A computational letter-perception model based on conceptual roles is currently being implemented by the first author (McGraw, 1992).



Figure 2: A graphical representation of some of the most standard sets of roles for the 26 lowercase letters. Only letters on the first line are shown with multiple sets of roles. There are other possible standard ways (not shown here) of conceptualizing many of the letters. Roles are not themselves shapes; instead, they are abstract sets of norms at the conceptual level into which parts of letterforms can be fit. These pictures are meant to show how such roles work together to produce letter-conceptualizations. The black dots shown represent between-role relations (r-roles).

interaction between roles also matters. For example, there might be a very strong post and a very strong bowl but the way they interact could make them form a bad ‘b’.)

The style of a letterform is a function of how the various roles in the letter-conceptualization are filled. Perhaps a more vivid and accurate way of saying this is that style has to do with *how norms are violated*. For instance, consider the role *crossbar*, which belongs to conceptualizations of ‘e’, ‘f’ and ‘t’. Possible norm violations for this role are many, but certainly include the following: “crossbar too high”, “crossbar tilted upwards”, “unusually short crossbar”, “crossbar missing”, etc. Any such violation is a stylistic hallmark that helps constitute a letterform’s style. In extreme cases, stylistic aspects of letterforms have clear effects on the ease of their categorization.

1.4 How roles are filled

Figure 3 shows how letterforms are comprised of parts that fill a letter-conceptualization’s roles. During perception, representations of such parts are formed under top-down pressure from roles and are sensitive to context. As stated in Palmer (1978, page 96), “components [of parts] enter into relationships with other components, resulting in larger structural units whose importance supersedes that of its constituents.” We hold that most of the “importance” attributed to the emerging parts stems directly from their role-filling ability. In other words, the way in which a part fills a role directly determines its “goodness.”

Sometimes norms associated with roles must be *violated* in order to accommodate a given letterform’s parts. In this sense, the roles are “plastic”. This plasticity is a result of allowing certain slots in the mental representation of a role to be filled by any of a range of closely related part-descriptions. The plasticity of norms associated with a role is context-sensitive and varies according to conceptual pressures brought to bear by the situation. Some of the norms making up the representation of a role are more malleable than others. We argue that the non-brittle nature of human letter perception is due to plasticity at the level of roles.

letter perception seems to have been largely overlooked.²

1.2 Parts versus roles

Palmer (1978) demonstrates the existence and utility of representations of structural units on a level above that of lines and/or points, yet below that of a whole figure. He claims that such structural units (similar to what we call *parts*) play a critical role in shape perception, strongly affecting categorization and similarity judgments. Palmer’s structural units are perceptually attractive because they have certain simple, “natural” attributes closely related to properties described by gestalt psychologists. These properties do not depend on a learned set of higher-level categories, and for that reason, Palmer’s structural units are built up in a wholly bottom-up manner by the perceptual system.

We readily agree with Palmer’s idea of intermediate-level structural units, or parts, but go a step further and argue that such parts are formed under a mixture of gestalt-like bottom-up pressures along with top-down pressures coming from conceptual-level category representations. Under our hypothesis, roles provide top-down pressures on the parsing of letterforms into sets of parts. The emergent properties of “good” parts will tend to fall within *norms* provided by roles.

The parsing of a letterform into parts can be understood by analogy with the parsing of words into speech sounds. Linguists and phoneticians have attempted to exhaustively list all the many possible sounds that humans can utter in any language. They call these sounds *phones*. Each language, however, uses a small selection of the possible speech sounds to distinguish words. These language-specific sets are called *phonemes*. The phonemes of different languages carve up the space of phones in diverse, often incompatible, ways. This means, for example, that while English speakers can readily distinguish the vowel phonemes of *beet* and *bit*, speakers of Greek or Spanish find this extremely difficult. When an utterance is perceived by a listener, the constituent phones are recognized only as belonging to certain phonemes of the listener’s language. In other words, one’s native language conditions one to hear certain types of speech-parts and not others. This is a quintessential example of top-down influence in perception. In much the same way, the parsing of letterforms into visual parts is influenced in a top-down way by the roles defined by the letter-conceptualizations or character-conceptualizations of the writing system one has grown up with.

Takano (1989) identifies four distinct and orthogonal types of information that are extracted from figures by the perceptual system. These include the identity (shape) of individual parts, orientations of individual parts, the connectedness of parts to other parts, and relative orientation between part pairs. The first two types of information Takano discusses are analogous to information contained in our roles. The second two types are analogous to information contained in r-roles.

1.3 Letter-conceptualizations as sets of roles

Some conceptualizations for a given letter category are better than others. A conceptualization’s quality depends on how strongly the spectrum of instances that it defines are rated by viewers as members of the intended category. Those conceptualizations that define the strongest instances are more important than their rivals, and provide *norms* for a given letter.

Figure 2 is a graphical representation of some, but by no means all, of the most standard sets of roles for each of the 26 lowercase roman letters. Roles and relationships between roles make up the internal structure of a letter category. Category membership at the whole-letter level is partially determined by category membership at the lower level of roles. (We say “partially” because the

²This omission is briefly mentioned in the surveys of Mantas (1986) and Gaillat and Berthod (1979).

1.1 What is the letter ‘a’?

Letterforms are subtler than people generally realize. Most literate people take letters for granted, not considering them deeply. For example, most people think the letter ‘a’ is just a *shape*. A closer look reveals an interconnected web of abstractions that make up the idea of ‘a’ itself. Figure 1 shows a number of very different lowercase ‘a’s, behind all of which lies a single highly abstract idea. The physical similarities of the shapes are minimal, but most are easily recognizable as ‘a’s.

The concept lowercase-‘a’ can be thought of as a marriage of two smaller ideas: (1) the idea of a curved umbrella-handle-like curve touching the baseline on the right, and (2) the idea of a smaller c-like curve nestled below the umbrella-handle. (See the first ‘a’ in Figure 2.) These two conceptual components, or *roles*, are not explicit shapes *per se* but are *ideas* supplying norms and constraints on what letter parts should be like, what their acceptable bounds are, how far they can be stretched before they no longer work as letter parts, and how they relate to each other to define an entire letterform.¹

In order to better distinguish the concept of a letter from the various geometric shapes that may instantiate it, we introduce some terminology. We distinguish three levels of abstraction, starting at a highly abstract level and becoming more concrete as we move towards the actual graphic letterform. A *letter-concept* is the abstract, shapeless notion of a letter, with which there is no particular style associated whatsoever. A *letter-conceptualization* is a breakdown of a letter-concept in terms of particular roles. For instance, one conceptualization of ‘b’ might state that ‘b’ consists of a *post* attached in two particular spots to a left-facing *open bowl* sitting to the right of the post, and on the baseline. A different conceptualization for ‘b’ might state that ‘b’ consists of a post attached in just a single spot to a *closed loop* sitting to its right on the baseline. These two conceptualizations, along with any number of other ways of thinking about what ‘b’ is, constitute the full letter-concept of ‘b’. A *letterform* is an actual graphic shape drawn on paper to realize a given letter. Letterforms have *parts* that fill the roles (either well or poorly) of a letter-conceptualization.

A vivid example of the shape/concept distinction involves lowercase ‘x’. For most people who learned to write in the United States, the letter-concept for ‘x’ consists of just a single conceptualization, whereby the letter is thought of as consisting of a forward slash and a backward slash of the same size, which cross somewhere near the middle. This conceptual breakdown remains the same even when one learns cursive writing. English and Italian schoolchildren, by contrast, are taught to draw a lowercase cursive ‘x’ as a pair of small crescents facing away from each other and “kissing” somewhere in the middle. If we look at a printed ‘x’ in this way, we are suddenly struck by this new conceptualization. The shape on the paper has not changed, of course, but our inner way of conceiving it is completely fresh and intriguing, and to anyone who is curious about letters, this new vision of ‘x’-ness implicitly but quite clearly makes a host of potential suggestions about *other* letters as well.

The conceptual pieces that are critical for ‘x’-ness include not just the slashes or crescents, but also such things as the location of the four *tips*, as well as the central *crossing point* or *kissing point*. We call these inter-role relational structures *r-roles*. Each role or r-role has a different degree of *importance* to the letter — the degree to which its presence or absence matters. Of course, different concrete shapes instantiate a given abstract role more strongly or weakly than others. In other words, roles, just like *wholes* (complete letters), are concepts in their own right, with somewhat nebulous boundaries. The difference is that membership in a role is easier to characterize than membership in a whole, so that reducing wholes to collections of interacting roles is a step forward in simplification. Schyns and Murphy (1994) and Palmer (1977) note the facilitating effect of similar perceptual hierarchies. Strangely, the idea of applying such perceptual hierarchies to machine-based

¹ Blesser et al. (1973) discuss a related idea.

1 Introduction

Literate humans are adept at recognizing letters in many diverse forms. Large variations in style — especially apparent in display faces for advertising — do not seem to hinder our ability to properly identify characters. Idiosyncratic characters like some of those displayed below are easily recognized even when encountered for the first time.

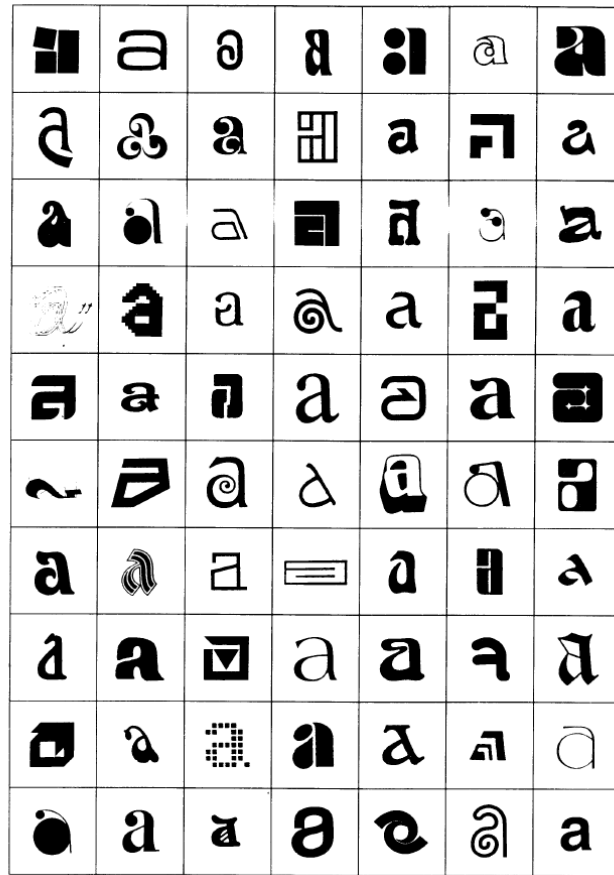


Figure 1: Seventy ‘a’s taken from typeface catalogues. What do they all have in common? What is the *idea* behind lowercase ‘a’?

Character recognition by people is still vastly superior to that of machines. Why is it that people are able to quickly and easily recognize letterforms while machine recognition systems fail? We believe what is required is an approach using abstract conceptual representations of letters. We argue that such high-level representations are found in the human mind and have clear effects on perceptual processing. We will describe such representations and empirically demonstrate their existence.

Abstract

We present the results of an experiment in letter recognition. Unlike most psychological studies of letter recognition, we include in our data set letters at the fringes of their categories and investigate the recognition of letters in diverse styles. We are interested in the relationship between the recognition of prototypical letters and the recognition of eccentric, highly-stylized letters. Our results provide empirical evidence for conceptual constituents of letter categories, called *roles*, which exert clear top-down influence on the segmentation of letterforms into structural components. The human data are analyzed in light of two computational models of letter perception — one connectionist and the other symbolic. Performance of the models is compared and contrasted with human performance using theoretical tools that shed light on processing. Results point in the direction of a model based on a role-based approach to letter perception.

Roles in Letter Perception:
Human data and computer models

Gary McGraw[†] John Rehling[†]
Robert Goldstone*

[†]Center for Research on Concepts and Cognition
Indiana University, Bloomington, Indiana 47405

&

Istituto per la Ricerca Scientifica e Tecnologica
Loc. Pantè di Povo, I-38100 Trento, Italia

gem@irst.it rehling@irst.it

*Department of Psychology
Indiana University, Bloomington, Indiana 47405
rgoldsto@ucs.indiana.edu

CRCC-TR 90

July 5, 1994