•5• *The role of knowledge*

Knowledge can affect structure

We have now seen that structural information can determine pragmatic subjects – it isn't all to do with lines and colours. Pop-out also occurs with items whose size, shape and colour are the same, but whose structures are different, as in Figure 5.1(a). The 'cube' that points the 'other way' stands out. You might think that this is just because it has been rotated, but there is no pop-out with similar objects that contain the same number of lines, angles, and so on, as in Figure 5.1(b).

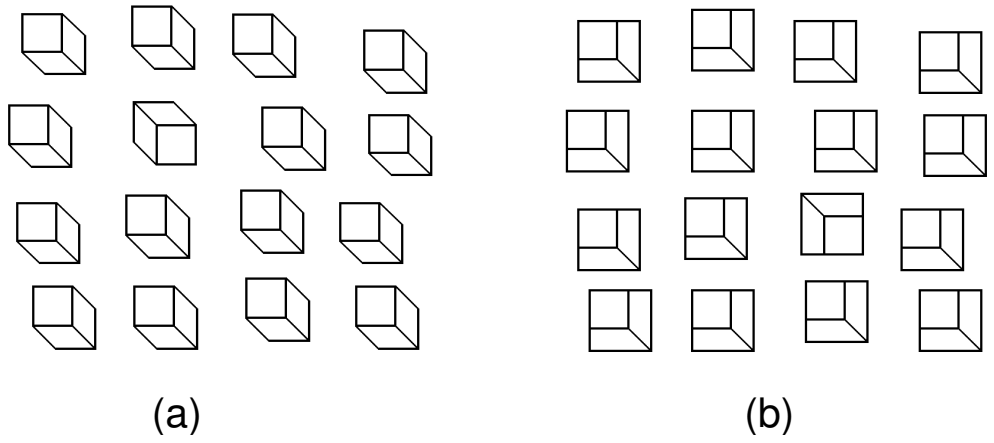


Figure 5.1: two very similar sets of objects with different pop-out effects

Pop-out for the objects in Figure 5.1(a) must be due their grouping in the way that is shown in Figure 5.2. Lack of pop-out for the objects in Figure 5.1(b) must be due to their forming a single group, which needs to be searched through for the 'different' object to be found.

This is a good example of how the propositional knowledge that a person has access to can affect how they perceive the display. If you are told that the objects in Figure 5.1(b) show the end of a megaphone, or an empty box 'descending' into the display (as in Figure 5.3), then you can form a mental image of a three-dimensional depth relationship between the lines on the screen. Nothing has changed visually, but now the one item 'facing' the other way pops out of the display.

Seeing it as three-dimensional has made its 'direction' obvious, and it has been grouped separately from the other 'empty boxes'. You have almost certainly seen lots of pictures of cubes like those in Figure 5.1(a), and so you were able to see them as three-dimensional straight away, but the use of perspective in the objects of Figure 5.1(b) is unconventional, and so you had to be given a hint as to how to interpret them as three-dimensional.

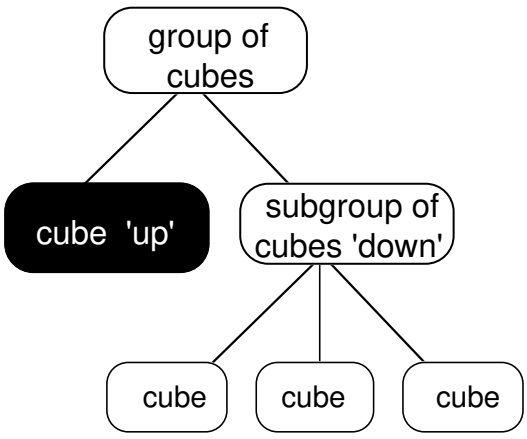


Figure 5.2: the structure diagram of the 3-D cubes.

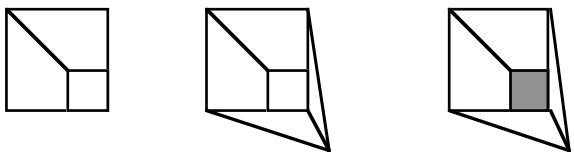


Figure 5.3 - reinterpreting the structure of the objects in Figure 5.1(b)

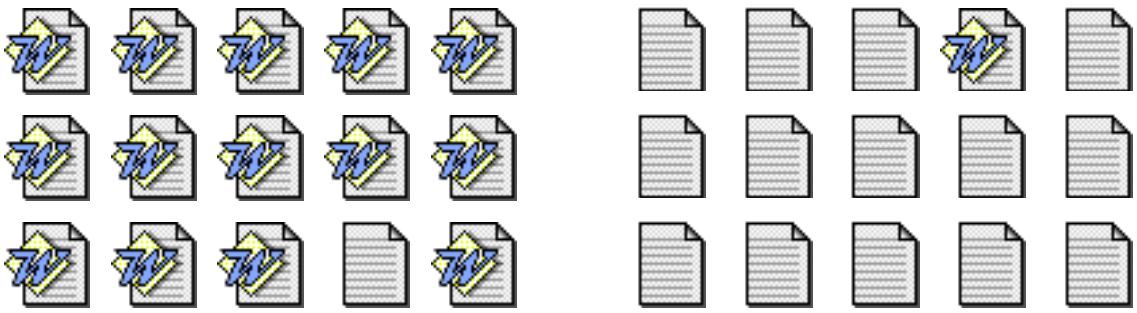
Exercise Set 5

1. Draw the structure diagram for Figure 5.1b, showing the representations formed without any propositional knowledge, so that objects are perceived as 2d.

2 Draw the structures with propositional input, so that they are perceived as 'empty boxes'. Which object is the pragmatic subject?

3. Look back to the structure diagrams that you drew for questions 2 and 3 in Exercise Set 4 (for the document icons). Which of the document icons is the most 'complex'?

4. Draw structure diagrams for these two arrays of icons, and identify which icon, if any, is the pragmatic subject:



Learning the meaning of objects

A computer user who was searching through the abstract and representational icon arrays of Figure 5.1 had to generate mental images of 'target' icons that they then compared the 'candidate icons against. Before the experiment could begin, the users had to learn what each of the icons 'meant' – propositional knowledge. When they were then asked to 'find the icon for *delete line*', they were able to retrieve their propositional knowledge and use it to generate an object representation of the icon that they were looking for, without any visual information.

In practice, users can never be given exhaustive training on computer programs, to make sure that they know exactly what every icon looks like and means, nor what the structure of every dialogue box is going to be. What users will learn about the structure of interface objects depends on how they use them, as was shown by two hypertext databases that used versions of the same display design, with one slight difference (Figure 5.4).

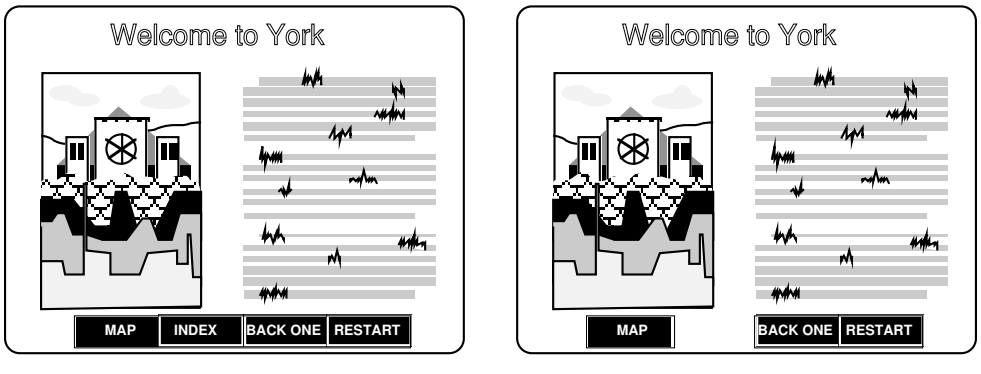


Figure 5.4: The two versions of the York Hypertext interface

The prototype 'visitors guide' had been built to let people read text and see pictures of York. They could navigate by clicking on 'hot spots' in the text, or by using a set of buttons at the foot of the screen. In Version A of the interface, these buttons allowed them to access a schematic 'map' of all of the screens, an alphabetic 'index' of the screens' titles,

to go 'back-one' to the previous screen they had seen (rather like an undo function), or 'restart' to go right back to the first screen.

Version B was the same, except that the 'index' button was omitted. People were shown how all of the functions worked, and then given some questions about York that they had to find the answers for in the hypertext. A typical pattern of exploration involved the users selecting a screen, and then realising that it didn't offer them any help, and so they would use the 'back-one' button to retrace their steps. Sometimes they would get completely lost, and 'restart'.

In Version A the navigation buttons were presented together as a block at the bottom of the screen, and since they had similar shapes and colour, they formed a group on the screen (Figure 5.5). To attend to any one of these buttons, people first had to attend to the group as a whole, and then make a transition into the group's structure.

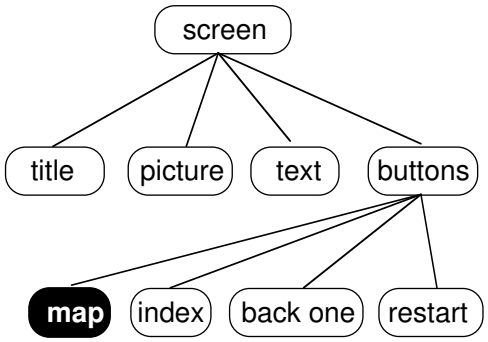


Figure 5.5: the structure of Version A, with the 'map' button forming the subject following a transition into the structure of the navigation buttons

When they did 'zoom in' to the structure of the groups, the individual objects were the four buttons, but since they were all identical (apart from the textual labels), none of them 'popped out' as a pragmatic subject according to shape or colour. In these circumstances, users would be most likely to scan across the buttons from left to right, reading the labels as they would normal text. A transition path that would be required to find the 'restart' button in this interface is shown in Figure 5.6.

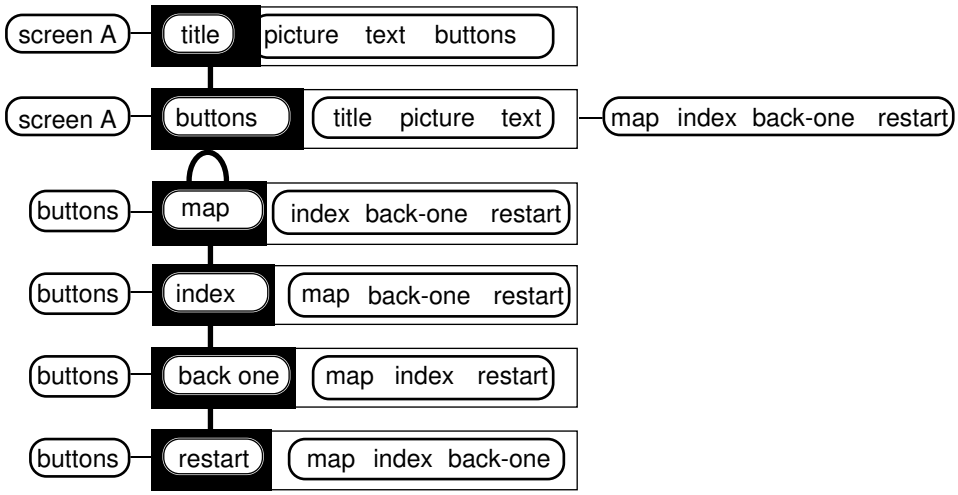


Figure 5.6: transitions required when using the buttons in Version A

After people had answered all of the test questions, they were asked about the various functions and buttons, and what they all did. Most of the people who had used Version A were found to understand that the map and index buttons could help them navigate around the system, even if they had not actually used them (they had, after all, been shown them in the introduction). Surprisingly, the people who had used version B were found to have less knowledge about the purpose and use of the 'map' button. A look at the structure of the interface shows why this might be (Figure 5.7).

In Version B, the gap left by the omission of the 'index' button breaks up the group of navigation buttons. Now the 'map' button stands on its own - and depending upon the size of the picture, might actually be associated structurally with it instead of with the other buttons. When people had to use 'back one' or 'restart', they no longer had to encounter the 'map' button (the transition path is shown in Figure 5.8). This meant that as they used the system, they did not encounter the 'map' button while they were navigating, and so the information they had been given at the start of the session about its function was not integrated into their propositional understanding of the system's functionality.

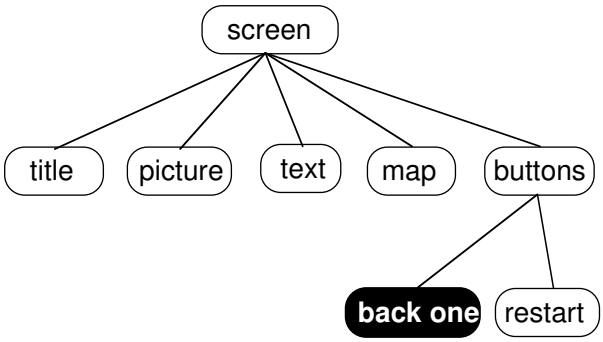


Figure 5.7: the structure of Version B, with the 'back one' button forming the subject following a transition into the structure of the navigation buttons.

The transition path diagram for Version B shows that it is indeed easier for the users to find and use the back-one and restart buttons, but it is at the cost of their understanding of the map button. A conclusion we can draw from this example is that grouping items together visually not only helps users to locate them *as a group* when they need to use them, but it also helps them to generalise about their common functionality.

Objects that are grouped together can come to share propositional meaning, with experience

Propositional knowledge about one of the group will tend to 'rub off' on the others. Of course, this can only be beneficial to users if there really is some similarity in the functionality of the grouped buttons. If the 'map' button had actually shown a geographical map of York, rather than of the hypertext, it would not have helped them to navigate around the system at all, and so it would have been misleading to place it with the other navigation buttons, even though they are all 'objects the user can press'.

Functionality is 'what the user does with a system', not 'what the system could do'

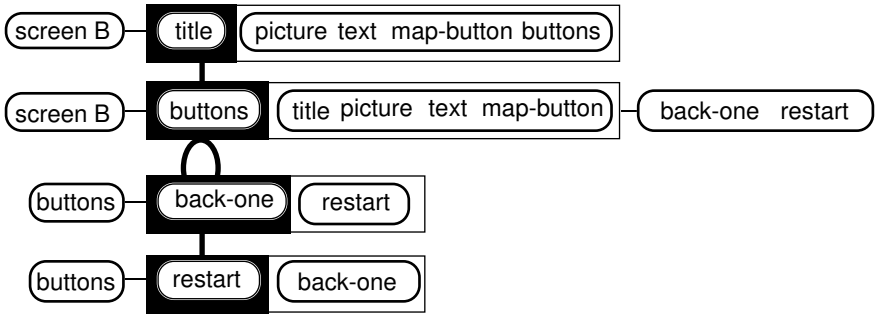


Figure 5.8: visual transitions when using buttons in version B

Competing groupings

The contribution of propositional knowledge to perception might seem to make the task of display design a whole lot easier: after all, if users can be told what to look for, and can learn how to group objects, why should it matter what the visual information is like? The pop-out examples shown earlier in this guide should convince you why this argument fails: for example, even when you know that all of the circles are just circles, a differently coloured circle still pops-out.

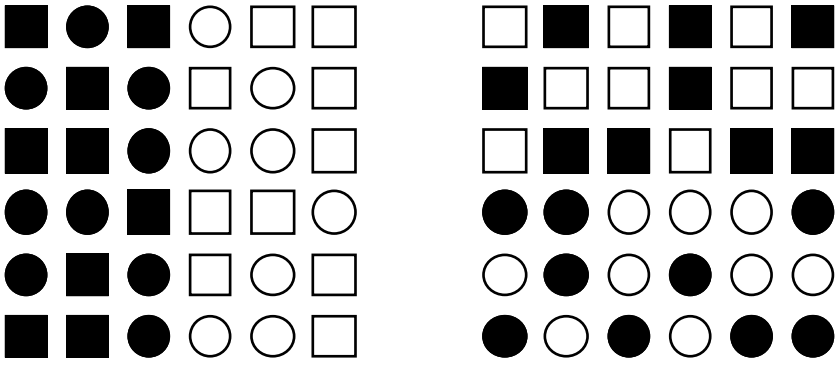


Figure 5.9: Colour dominates shape

Those examples were selected especially because they showed the effects of pop-out very strongly, of course. They were generally very simple, with only one attribute changing to influence grouping. Even in more complex arrangements, the dominance of the visual contribution to object representations can be just as convincing.

The left hand array in Figure 5.9 shows again how strongly colour can determine the grouping of objects. It is easy to see this as a group of black objects and a group of white objects, even though the objects themselves have different shapes. In the right hand array, where the objects are arranged by shape, it is not so easy to see two distinct groups.

Here the randomness of the colour dominates the orderliness of shape, preventing the visual-to-object transformation forming a pragmatic subject in the structure of the array. Figure 5.10 shows structure diagrams of the two arrays. In the left hand array, there is an 'intermediate' level of grouping between the array and the individual objects, but this is missing in the right hand array. Once you 'know', propositionally, that there are two 'shape' groups in the right hand array, you can impose this structure on the object representation, but it seems to require continual mental effort to do so. As soon as you look away and back, the randomness of the colours dominates once again.

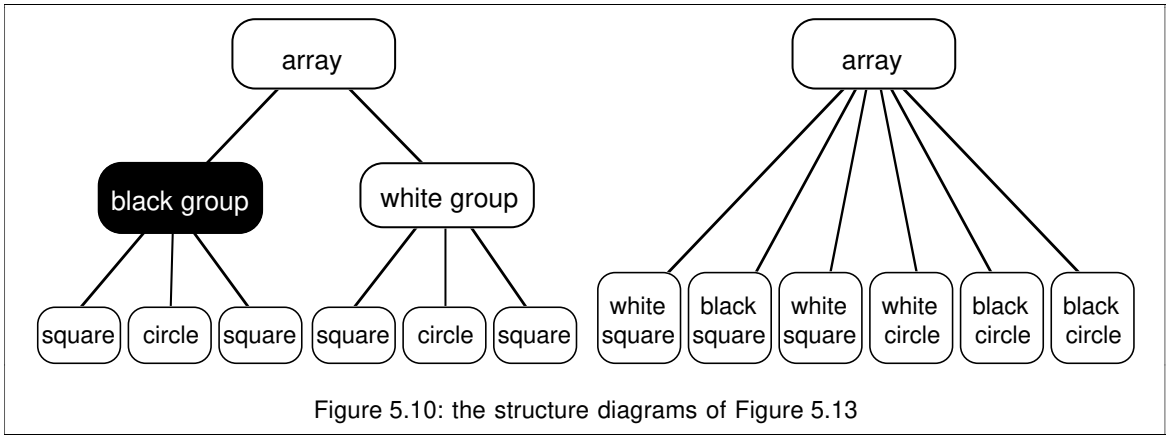


Figure 5.10: the structure diagrams of Figure 5.13

The arrays (a) and (b) in Figure 5.11 show pop-out due to the shape of the oblongs. In both (a) and (b) the object in the array that does not have the same shape as the others becomes the pragmatic subject and pops out. In (c) one object differs in colour, and the pop-out effect is enhanced. As in Figure 5.9, colour is stronger than shape, for in (d) the white, horizontal object pops-out, while the black, vertical form does not. This is despite the fact that the white object has the same shape as the black objects, while the black object has a different shape: so colour dominates shape.

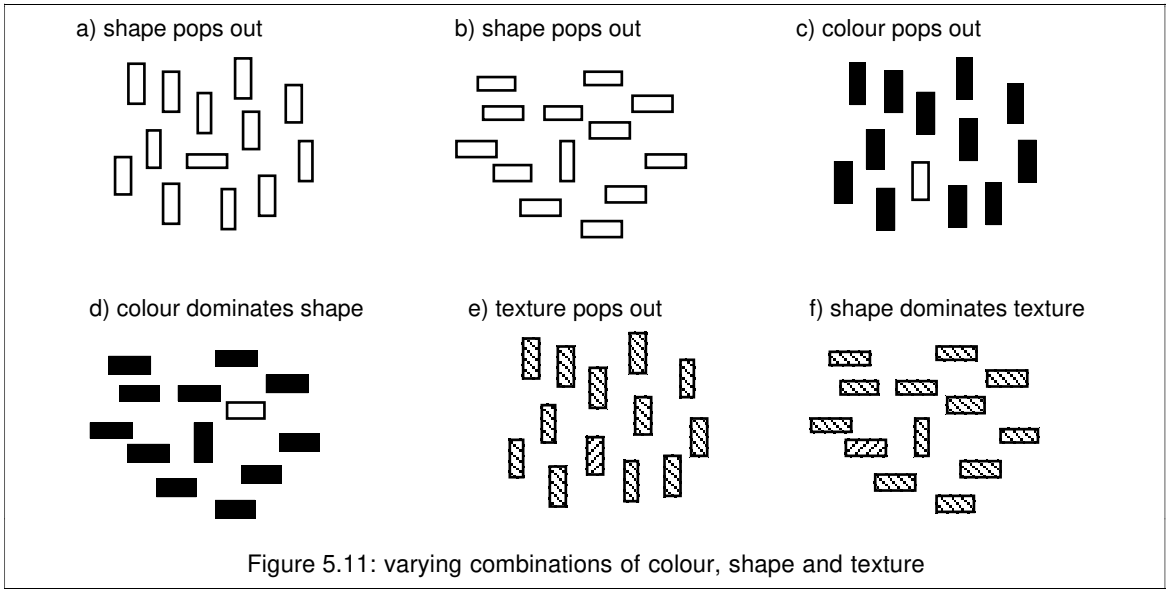


Figure 5.11: varying combinations of colour, shape and texture

Arrays (e) and (f) introduce a new attribute, texture. Texture is defined as a regularly spaced, repetitive pattern where each of the objects of the pattern is individually perceptible, but where the objects are much smaller than the whole that they fill. Although it is common to think of texture and colour as very similar, since they appear to be properties of the surface of objects, visually they have quite different properties. Unlike colour, texture does not dominate shape.

In (e) the oblong with a different texture (direction of stripes) to the other oblongs does pop out weakly, but in (f), the upright oblong that has the same direction stripes as the other oblongs stands out instead. There is a horizontal, differently textured oblong in the array, but it is much harder to locate now than in (e). Clearly the 'difference' of the texture is not as influential as that of colour. Like size, brightness and contrast, texture is also a 'depth cue', with objects of a finer texture appearing further away than objects of a coarse texture (Figure 5.12).

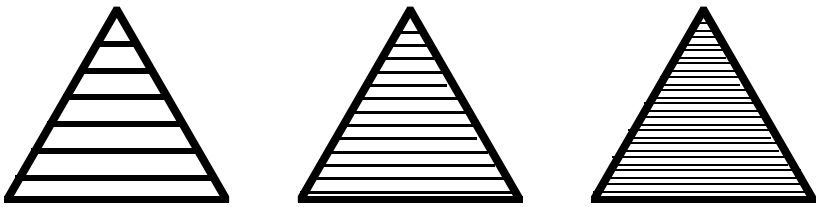


Figure 5.12: objects with coarse texture appear closer than objects with fine texture.

A coarse texture, like a larger size and increased brightness, makes an object look **close.**

Although the textures in Figure 5.11 were differently sloped, they were equally bright, and so they did not suggest that any of the oblongs were any nearer or further away than any of the others. This meant that the texture attributes did not force them to join separate groups, allowing the shape to dominate.

In grouping objects, colour and shape ('difference') dominate closeness.

The grouping of objects on the basis of attributes like colour, size and shape is such a pervasive part of design that we tend to take it for granted in many circumstances. Text is a good example. Letters that are written in the same font and face are easily grouped into words, even if they do not actually spell recognisable words. Changing the attributes of letters makes the words much harder to read, because their component objects become more 'visible' than the whole word (Figure 5.13), even though the word boundaries (spaces) are still there.

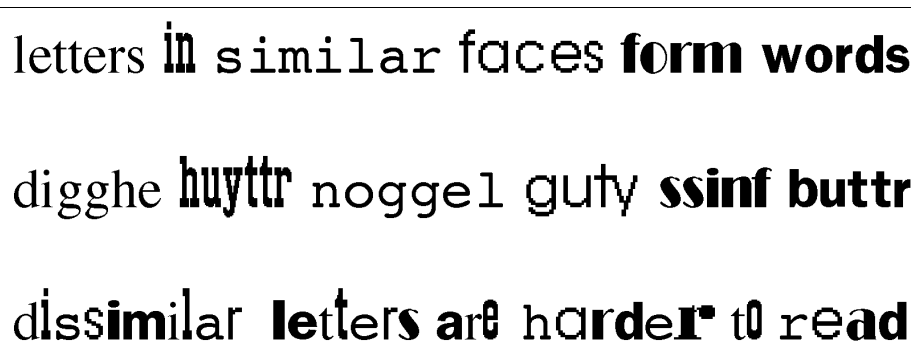


Figure 5.13: similar attributes of letters help them form easily readable words, even if they are nonsense, but letters with dissimilar attributes are harder to group into words, even if they make sense

Exercise Set 6

1. Draw a structure diagram for parts c, d, e and f of Figure 5.11.
2. Identify the pragmatic subject of the main group in each diagram.
3. Which of the subgroups also have pragmatic subjects?
4. According to the answers that you have given for question 3, is it easier to locate the differently shaped oblong in part d, or the differently textured icon in part f?