How is the alphabet stored?

Using priming to distinguish direct association from serial search

Liesbeth Flobbe
Supervisor: Hedderik van Rijn

Rijksuniversiteit Groningen

February 28, 2005

Abstract

Alphabetic retrieval is a prototypical task that is studied to gain insight into how humans learn and process long lists. We shall study two conflicting models of this process: serial search and direct association. To distinguish between these models, we shall derive predictions about priming effects that occur when items are paired. In a new experiment, we measure these priming effects. Although the small data set does not allow strong conclusions, it shows that a pure associational model alone is too simplistic.

How is the alphabet stored? How do people retrieve letters from the alphabet? Different accounts of how humans store and access the alphabet, or other long lists with little explicit structure, have been proposed. A good model must be able to explain human performance, and especially reaction times (RTs), in experimental tasks.

Tasks that have been studied in experiments include: reciting the alphabet from a specific letter, saying the next letter, judging whether two letters are in the correct alphabetic order, etc. All these experiments have found an increase in reaction times towards the end of the alphabet, as well as a distinctive pattern of peaks and valleys across the alphabet. In this paper we shall focus on this alphabetic retrieval task: A letter (the probe) is presented visually, and the subject has to say either the following or preceding letter in the alphabet. In the forward condition, the subject has to say the next letter in the alphabet. In the backward condition, the subject has to say the preceding letter.

A pattern relating to this task is shown in Figure 1. Note how the location of peaks and valleys is consistent between the forward and backward tasks.

1 Models of alphabetic retrieval: serial search vs. direct association

Klahr, Chase, and Lovelace (1983) propose a serial search-model of alphabetic retrieval. To find the letter following or preceding a probed letter, the subject has to 'recite' the alphabet from a specific 'entry point' until the probe letter is found (or one further to find the answer, in the forward search task). The reaction time depends on the time needed to find the entry point and the number of steps from the entry point to the probe letter.

According to the direct association model of Scharroo, Leeuwenberg, Stalmeier, and Vos (1994a), no serial search is necessary. Letters have direct associations with their successors, and the strength of this association determines the reaction time.

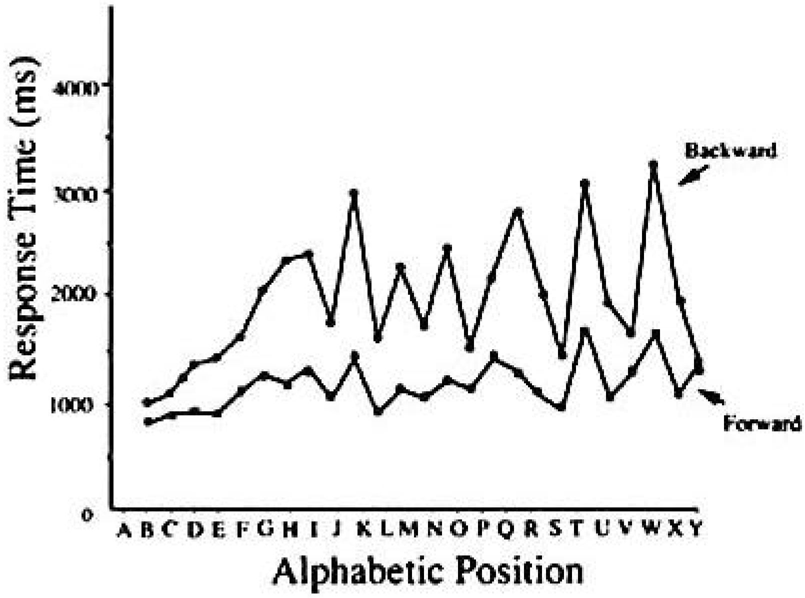

Figure 1: Reaction times (Scharroo et al. 1994a)

Forward vs. backward search

The model of Klahr et al. (1983) applies to both forward and backward searching. Scharroo et al. (1994a) leave open the possibility of serial search in the backward condition, while rejecting serial search in the forward condition, because the alphabet is learnt in the forward direction only, and direct associations with predecessors might not be available. However they also state that their experiment does not support the serial search model even for the backward condition, and that the Klahr et al. model has little value in explaining their results. So their position on serial search in the backward condition is not entirely clear.

A reply to Scharroo et al.'s work (Klahr 1994) proposes that a new model should be developed, which should combine both the serial search and the direct association model. If a sufficiently strong association between letters is available, this association is used; otherwise a serial search is performed. The article does not specify when such a direct association will be available, but the distinction between the forward and backward tasks seems a plausible candidate.

However, in Scharroo's rejoinder (Scharroo 1994b), she states she sees little use in such an arbitrary combination of models. A pure associational model is sufficient to explain the data, and a serial search component has little to add. The position in this article seems more radical than in the 1994a article because even in the backward search task it does not allow for a serial search process. Unfortunately, no account is given of how people learn backward associations between letters. Experiments have consistently shown higher reaction times in the backward task than in the forward task, which implies that a backward association is weaker than a forward association.

Chunks

According to Klahr and others who think humans use a list-structure to store the alphabet, the alphabet cannot be learnt directly, because it exceeds the capacity of working memory. The different subgroups in which the alphabet is divided during learning, and also during subsequent storage, are called chunks. When a chunk boundary must be crossed to find the answer to a test item, this results in significantly longer reaction times. To Klahr et al., chunks are also the preferred entry points for initiating a serial search: a search will always start from the first letter of a chunk.

To Scharroo et al., a chunk is "just a series of letters with strong associations, enclosed between weak associations" (Scharroo et al. 1994a, p. 239).

Individual differences

In Klahr's experiments with American subjects, he finds a strong interpersonal agreement on chunk boundaries. This segmentation coincides with the phrasing of the nursery song through which the alphabet is taught in American schools. Scharroo et al. however, in their experiment with Dutch subjects, find larger differences between subjects. They argue that this probably reflects the absence of a common method to teach the alphabet in the Netherlands. In both experiments interpersonal agreement on chunk boundaries decreases towards the end of the alphabet and chunk sizes towards the end of the alphabet are smaller.

Increasing RTs across the alphabet

Overall reaction times increase towards the end of the alphabet, and so do the RTs at the local minima that, in the serial search model, represent the beginning of chunks. According to Klahr et al., this increase in local minima occurs because access to entry points is slower for chunks later in the alphabet. In their account, this is explained by a serial search through all chunks to find the chunk containing the probe letter, which precedes the search within the chunk.

Scharroo et al.'s model (1994a) does not model increasing RTs at all, although in the 1994b article a parameter is added for this. They state that the overall RT increase is due to a primacy effect: the beginning of the alphabet has been repeated more often, therefore the associations between the letters are stronger at the beginning. They do not find an increase in local minima in the results of individual test subjects, rather they claim that the increase in the aggregate data is a result of averaging. Because the chunks are smaller towards the end of the alphabet and because variability between persons is greater, averaging results in increasing local minima.

Although we will have to take into account this increase in RTs across the alphabet, my experiment is not designed to decide between different explanations for this increase. We will focus on (possible) serial search within chunks only.

Predictions for priming

Given the difference between American and Dutch subjects, it is hard to decide which model fits the experimental data better. Therefore, we will derive new predictions about how priming can influence RTs. The results might help decide which model is correct.

The task is the same as described earlier: the subject is presented a letter and has to say either the next or the preceding letter in the alphabet. However, items will be paired to form prime-target combinations. For convenience, we will always refer to the first item of such a combination as the prime, regardless of whether we think this item causes priming or not.

An example would be the combination $D-, F-$. The prime item is $D-$ (the $-$ indicating that the task is to say the letter before the D) followed by a target item $F-$. The RT on this target item is compared to the RT on the same target item when preceded by an item $O-$. If the RT on the target item is significantly faster for the first combination than for the second, we can say that the $D-$ item somehow primes the $F-$ item.

We will distinguish three models, based on the described literature. For each model we will describe what predictions for priming can be derived from it. The examples assumes that the letters A to F are all in the same chunk.

SS (strong serial search)

Always serial search, both in the forward and backward condition.

This corresponds with the Klahr. et al (1983) model.

A prime item $C+$ or $D-$ will always cause someone to 'recite' from the beginning of the chunk until the prime is reached (it doesn't matter whether the next or the preceding letter is asked): "A, B, C, D", assuming the chunk starts at A. This will activate all the letters from A to D.

For a subsequent target $F-$, the subject will need to search the series A to F. However, this search should be faster because many of the letters have been activated. The right entry point (rather trivial in this case: A) should also be found faster because it is still active. We could even argue that the search doesn't have to start at A, but can start where the preceding search left of, at D. Whatever the precise mechanism, we expect a priming effect, both when the prime item is $+$ and when it is $-$.

If there is a chunk boundary between prime and target, no priming can occur. But averaged over all letters of the alphabet, we still expect a priming effect.

DA (direct association)

Always direct association, both in the forward and in the backward condition.

This corresponds with the Scharroo et al. model. Although they claim to find a serial search in the backward condition plausible (1994a), this is not incorporated in the formal model (Scharroo et al. 1994a). Scharroo later takes the position that a combination of models adds no explanatory leverage (Scharroo 1994b). When we refer to DA, we mean a pure associational model.

To find the letter preceding or following the prime, only the association between these two letters needs to be activated. This will not effect the subsequent target item, unless the target item or its answer is identical to one of these activated letters. Therefore, there is no priming except identity priming (i.e. a prime and target are identical, or ask for identical answers).

FABS (forward association, backward search)

A simple combination of both models. To find the next letter, direct association is used. To find a preceding letter, a forward serial search is initiated. The entry point for this serial search is the beginning of a chunk.

If the prime item demands a serial search (in the backward condition) the subsequent forward associations will be primed. This priming will affect the RT of the target

	priming	**no priming**
prime −	D− F−	P− F−
prime +	C+ F−	P+ F−

Table 1: Conditions: example

item if it is in the backward condition, by the same reasoning as for SS. It will not affect the RT of the target item if it is in the forward condition (at least not if the prime preceded the target in the alphabetic order), since the forward task does not involve a serial search.

If the prime item is in the forward condition, only the direct association between the prime and its following letter is activated. If the target is in the forward condition too, our expectations are the same as for direct association. If the target is backward, the activated association would slightly speed up the serial search, if this association is part of the series being searched (which is the case if the prime preceeds the target in the alphabet).

2 Experiment

Item design

Because Klahr himself has proposed a hybrid model, our design does not test all possible circumstances in which priming can occur according to SS. Rather, it tries to distinguish between pure association and any form of search (SS or FABS). Therefore, the target is always asked backward. The prime can be both forward and backward.

This leads to a matrix of four conditions. Table 1 gives an example of each condition, with all examples using the same target.

The conditions always use the same distance between prime and target, as explained below:

no priming, prime − (*np*−) : the 'prime' is the 10^{th} letter after the target (if the target is between B and P), or the 15^{th} letter before the target (if the target is between P and Z). Because this distance is larger than any proposed chunk size, there can be no priming effect.

no priming, prime + (*np*+) : the same as *np*−, but this time the prime is +.

priming, prime − (*p*−) : the prime is the 2^{nd} letter before the target. This is the minimum distance needed to ensure that the answer to the target does not overlap with the prime (either the prime letter itself or its answer).

priming, prime + (*p*+) : the prime is the 3^{rd} letter before the target. Again, this distance is necessary to prevent overlap between prime and target. Note that for the same target in conditions *p*− and *p*+, the prime involves the same pair of letters (but which letter is the question and which is the answer differs).

Using these distances, we generated prime-target pairs for every target from $B-$ to $Z-$ for the no-priming conditions and from $D-$ to $Z-$ for the priming conditions. To these items, fillers were added to achieve the following checks and balances:

1. the $+$ and $-$ operator occur equally often for each letter (except A and Z),
2. sequences of the same operator (at most three in a row) occur equally often for each operator,
3. in the $p+$ and $p-$ conditions, the prime is never primed itself.

We organised our items with fillers in sequences of 3 or 4 letters. The sequences could be reordered without violating the third condition. Every subject received a different, random ordering of sequences.

Predictions for our 4 conditions

It should be obvious that we cannot assume that a $+-$ and a $--$ combination will have the same RTs on the second item. Therefore, a direct comparison between $np-$ and $np+$, and between $p-$ and $p+$ is problematic. There are three different possibilies:

1. If there is no priming, the previous operator does not influence performance on the next operator. (If there is priming, the previous operator might influence performance, in so far as different operators cause different search processes.)
2. If there is no priming, performance on the target will be slower if the subject has to switch to a different task (i.e. a different operator). Therefore, $np-$ is faster than $np+$.
3. If there is no priming, slow performance on the prime will spill over as slow performance on the target. Since $-$ is slower than $+$, performance on the target will be slower for $np-$ than for $np+$.

We can compare $np-$ and $np+$ to get an idea of the size and direction of the previous operator influence. We can then use this to correct the RTs for $p-$ and $p+$.

Assuming that there is no previous operator influence, the different models would make the following predictions on the rank order of the conditions, where $>$ means 'higher target RT / slower' and $<$ means 'lower target RT / faster':

DA:	$np- = np+ = p- = p+$
FABS:	$p- < p+ < (np- = np+)$
SS:	$(p- = p+) < (np- = np+)$

Assuming nothing about the previous operator influence, not even that its direction is consistent across priming and non priming conditions, we can only predict a partial rank ordering:

DA:	$np- = p-, np+ = p+$
FABS:	$p- < np-, p+ < np+$
SS:	$p- < np-, p+ < np+$

The differences between SS and FABS in these predictions are very minor, as we have not added items with a forward target.

Method

The subjects were 15 psychology undergraduates, participating for course credit. They youngest was 18 and the oldest was 24. There were 8 females and 7 males. 12 subjects spoke Dutch as a child both at home and at primary school. One subject spoke Frisian at home and Dutch at primary school. One subject spoke German both at home and at primary school.

The items were presented on a computer screen. After the subject pressed the space bar to start each trial, a + or − sign was shown for 0.5 seconds at the center of the screen, then the operator disappeared and a capital letter was shown at the same location. Subjects were to press the spacebar as soon as they knew the answer. They then were shown a question mark and had to type the answer. By letting subjects press the spacebar before typing the answer, we aimed to prevent a confounding influence from the different letter positions on the computer keyboard. Subjects were instructed to use only their index fingers, so movements had to be sequential. To discourage subjects from pressing the space bar prematurely, the question mark would disappear after 2 seconds. Subjects received no feedback on the correctness of their response, but they knew the response was being recorded.

The experiment took about 4 x 10 minutes. Subjects were offered a break at three times during the experiment, and were free to determine the duration of the break.

3 Results

One subject was excluded from our analyses because he had a remarkably high error rate (18% overall, but 30% on − operator). Because we required for our analyses of priming that both the prime and the target are correct, half of the data for this subject was unusable.

For the remaining subjects, the error rate varied from 1.7% to 9.5% overall, with a mean of 6.8%. For the − operator alone, the error rate varied from 2.0% to 17.6%, with a mean of 10.9%.

Since these error rates are rather high, we have looked into possible causes of these errors. For 62.8% of errors, the response given was actually a correct response, but for the wrong operator. Subjects never saw the operator and the letter at the same time, and this appears to have caused many errors. For another 15.5% of errors, no response was given within 2 seconds. Whether this is because the subject wasn't fast enough to type the answer, or because he forgot the operator and decided not the respond, we don't know. For 12.5% of errors, the response was two letters away from the presented letter, instead of just one. For the remaining errors, either the presented letter was repeated as the response, or a response was given that had so little to do with the question that we assume it was a typing mistake.

Items with reaction times of less than 0.3 seconds or more than 10 seconds have been filtered out.

We have analysed reaction times per item for all items (including fillers), without looking at priming yet. Figure 2 shows the reaction time (averaged over all subjects) for each letter. The solid line represents the forward task, while the dashed line represents the backward task. Letter position 1 represents $A+$ and $B-$, while position 25 represents $Y+$ and $Z-$. This alignment best shows the correspondence of peak and valleys between the two tasks.

Figure 3 shows 2 graphs of individual subjects. These figures illustrate the large

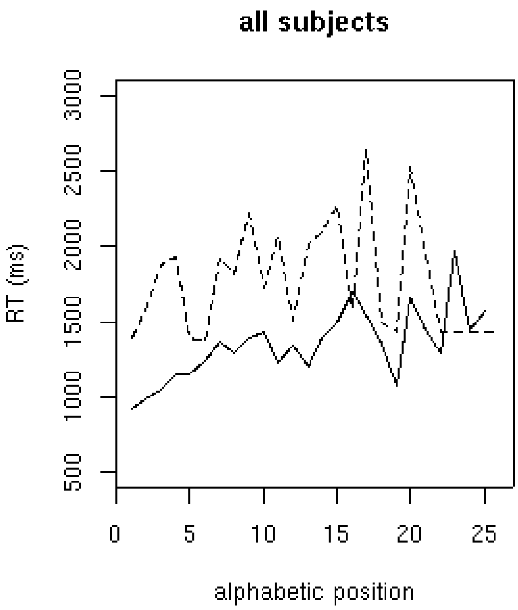

Figure 2: Reaction times per letter

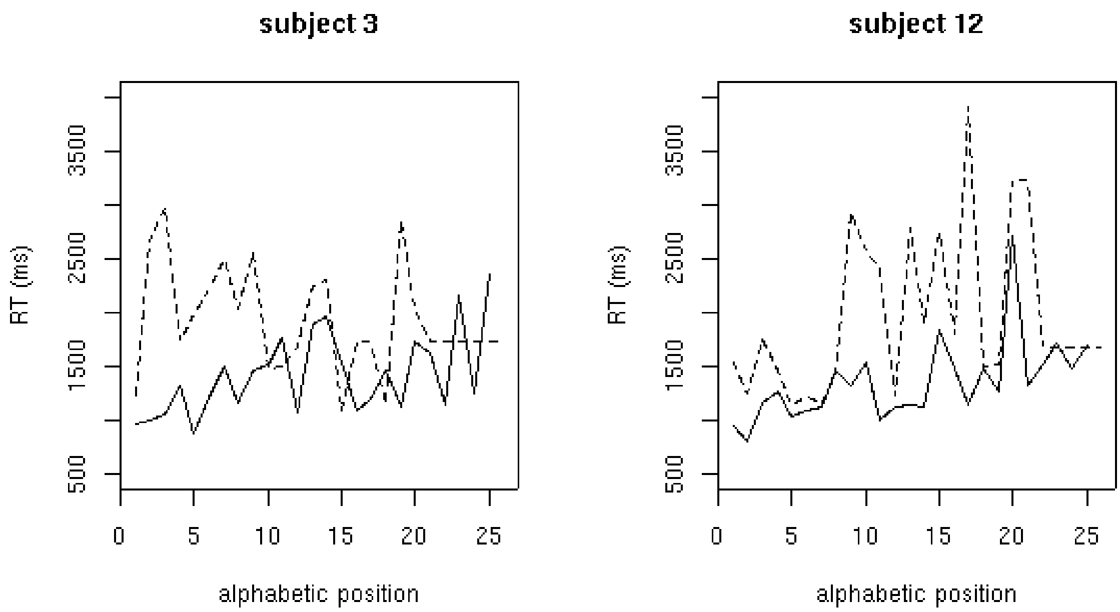

Figure 3: Reaction times per letter, individual subjects

$np+$	$p-$	$np-$	$p+$
1749 ms	1772 ms	1832 ms	1833 ms

Table 2: Average RT per condition

individual differences between subjects. Our averaged figure looks less smooth than the Scharroo et al. (1994a) graph that we reproduced in figure 1, but Scharroo et al. used more subjects (40). We think our averaged figure is consistest with the effects described in literature, especially with respect to the pattern of peaks and valleys and the congruence between the forward and backward tasks. The individual differences we find are not out of line with Scharroo et al. (1994a), who used Dutch subjects as we did. We cannot compare with Klahr et al. (1983) because they did not show individual results.

To analyse the effect of priming, we looked at the reaction time of the target letter as a function of the condition. The (intersubject) average per condition is shown in Table 2. Note that $p- < np-$, but also that $p+ > np+$, which does not match any of the (partial) rank orderings predicted earlier. The direction of the previous operator effect, with $p- < np-$, but $p+ > np+$, is not consistent. The differences are not significant, however. If the differences were significant, they would indicate an interaction between previous operator and priming, that causes priming to be *slower* than non-priming for the $+$ operator.

We used the statistical package R to create a linear mixed effect model of the data. The variable to be explained was the logarithm of the reaction time. The dependent variables were:

- The sequence number of the item in the experiment. This lets us model the learning that occurs during the experiment.
- The position of the letter in the alphabet, encoded as a factor.
- Priming: true in the $p+$ and $p-$ conditions.
- The operator of the previous letter.
- All two-way interactions between priming, previous operator, and sequence number.
- The subject. For every subject, a distinct error stratum was used.

We then stepped through the possible simplifications of this model to find the model with the lowest AIC value. This model contains the dependent variables sequence number, letter position, previous operator, and an interaction between previous operator and sequence number. As expected, there was a negative correlation between sequence number and reaction time, indicating a learning effect during the experiment. The interaction between previous operator and sequence number means that there is more learning when the previous operator is $-$ than when it is $+$. An ANOVA-analysis of this model showed that sequence number, letter position, and the interaction between previous operator and sequence were all highly significant at the $p < 0.001$ level. The previous operator alone was not significant, however ($p = 0.3254$).

Our computer model does not include priming: priming does not help explain the reaction times better.

4 Discussion

We have not been able to find a significant effect of priming. However, the conclusion that there is no priming is not warranted. The effect of the previous operator is not significant either, even though it is included in the model with the best AIC-value, and an interaction with this effect *is* significant. Because of the pattern of peaks and valleys across the alphabet, it was necessary to treat the letter position as a factor, instead of as a continuous variable. This means that the data is modelled per letter, per condition, per subject, which requires a very large data set.

We think that further research with a larger subject pool is useful. Such further research should also review the item design, to prevent correlations between priming and other possible factors as much as possible.

Our experiment has shown that using a computer keyboard as input device gives results comparable to using a voice key. This means experiments can be conducted with standard computer hardware.

We think it is prudent for future research using this alphabetic retrieval task, even if priming is not its object, to control for possible priming and for the previous operator.

References

[1] David Klahr, William G. Chase, and Eugene A. Lovelace (1983) Structure and Process in Alphabetic Retrieval. *Journal of Experimental Psychology*, 9 (3), 462-477.

[2] Jackie Scharroo, Emanuel Leeuwenberg, Peep F. M. Stalmeier, and Piet G. Vos (1994) Alphabetic Search: Comment on Klahr, Chase, and Lovelace (1983). *Journal of Experimental Psychology*, 20 (1), 236-244.

[3] David Klahr (1994) Plausible Models of Alphabetic Search: Reply to Scharroo, Leeuwenberg, Stalmeier, and Vos (1994). *Journal of Experimental Psychology*, 20 (1), 245-249.

[4] Jackie Scharroo (1994) Modeling Alphabetic Retrieval: Rejoinder to Klahr (1994). *Journal of Experimental Psychology*, 20 (2), 492-495.