RACE and the influence of timing on the human decision process

Lennart van Luijk

April 2009

Master Thesis
Artificial Intelligence
Dept. of Artificial Intelligence
University of Groningen, The Netherlands

Supervisors:
L. van Maanen (Artificial Intelligence, University of Groningen)
Dr. D.H. van Rijn (Dept. of Psychology, University of Groningen)

Summary

When a person is handed a simple question and is simultaneously asked to press a button after 4 seconds without counting, does the extra workload have influence on the reaction time or the performance on the question? Cognitive science is a field of research which deals with such questions, trying to explain cognitive processes in the human brain, such as decision processes. We studied the influence of timing on decision processes by combining a timing experiment (TE) with a lexical decision experiment (LD). In LD a string of letters is presented and the participant decides whether this is an existing word. We developed an ACT-R model of LD, and improved this with abilities to match more complex empirical data by making use of RACE (Van Maanen & Van Rijn, 2007), so that retrievals from declarative memory are not bound by limitations from ACT-R. From our model, combined with research which suggests that internal time perception is non-linear (van Rijn & Taatgen, 2008), we predicted that performance on a combined LD and TE task is dependent on the time at which the LD stimulus is offered during the time interval. Our models and the results from this experiment will be discussed.

Table of contents

1 Introduction

1.1 *Introduction*

When a person is handed a simple yes or no question and is asked to count until 5 while answering this question, how would the performance on answering the question be influenced? Certainly the individual would feel that it is harder to focus on two tasks at the same time, especially when there is a limited time for response. Will this be visible in the accuracy of the answers on the question? These are interesting questions to ask when trying to understand the human decision process, since a large part of the actions one performs are a result of a cognitive decision. Therefore, this process plays an important role in understanding the way the human brain works.

1.2 *Cognitive modelling and decision processes*

The human decision process is a prominent subject in the research field of Cognitive Modelling. When modelling a typical example of this process, one of the most widely used tasks is lexical decision. Lexical decision (LD) is a task in which a participant observes a letter string and has to decide whether this string is a genuine word or not and the reaction times are measured. There are models already available which model the LD process well (e.g. Wagenmakers, Ratcliff, Gomez, & McKoon, 2008).

The LD task can be simulated in a cognitive architecture (CA), which can be defined as '[..] a specification of the structure of the brain at a level of abstraction that explains how it achieves the function of the mind' (Anderson, 2007). There are several cognitive architectures, such as EPIC (Meyer & Kieras, 1997), Soar (Newell, 1990) and CLARION (Sun, 2006). We will use ACT-R (Anderson, 2007; Anderson et al., 2004), since we will model the decision process where retrievals from declarative memory play a role. ACT-R has a declarative module that can be adapted to our needs, making it the most suitable cognitive architecture for our model since the other mentioned CA's don't have this possibility. ACT-R is already able to explain empirical data from experiments that deal with declarative memory, such as picture-word interference experiments (L. Van Maanen & Van Rijn, 2008). Therefore, we will model the LD task in ACT-R.

Decision processes such as the LD task can be influenced by adding a second task, to study the influence on performance on the first task. An example of an interesting task to add is timing, since this can influence the LD task in different ways. For example, by adding a second task performance on the LD task may degrade, since the participant has the same amount of time to perform more actions than in LD only tasks. However, the way humans perceive time can influence the results as well. There is research providing evidence that the internal perception of time is non-linear (Van Rijn & Taatgen, 2008), which we can test in an experiment with a combined timing and lexical decision task.

In the LD task we will model, there are aspects that ACT-R is not able to explain, which we will describe in detail in the next chapter.
Modelling this is in theory possible with an alternative model for retrievals from declarative memory, called RACE (Van Maanen & Van Rijn, 2007), which is designed to explain the fine-grained level of the retrieval process. RACE should already be able to simulate simple LD tasks, since it is designed to be backwards compatible with ACT-R, and will be extended to model more complex LD tasks as well. We will then combine the LD task with a time estimation (TE) task, and design an experiment to study the influence of performing a simultaneous TE task on the performance at the LD task.

1.3 Theoretical background of ACT-R

1.3.1 ACT-R Introduction

ACT-R is a hybrid cognitive architecture in which a sequence of production rule executions describes behaviour in a task. Production rules implement procedural knowledge in ACT-R. Given certain conditions, these rules specify which actions to execute. For the execution of a production rule, the conditions are matched against the current information state. This state is represented by a set of buffers, each belonging to one of the specialized modules in Figure 1.1. Each module can have one or more buffers, which are the interfaces of the modules for information exchange with the other modules. The production rules can interact with these buffers by reading from them and writing information into the buffers. This interaction can take place simultaneously with several buffers, so that the modules can process tasks in a parallel way.

Each module processes one kind of information. For instance, the motor module executes motor commands. The imaginal and goal modules keep track of (sub) goals and intentions. The visual module handles visual perception, whereas the aural module handles auditory perception. The speech module handles speech output, and the declarative module is used for storing and retrieving declarative knowledge in memory. This knowledge (facts) is stored as chunks. This research will focus on the latter module.

The production rule system connects these modules, where each can be regarded as a theory on that particular aspect of cognition, to account for overall behaviour.

For a task such as lexical decision, the visual module is used to read the stimulus, the declarative module is used for recognition of the stimulus and the motor module controls the answer on the keyboard. The goal module may be used as well to keep track of the higher order goal, but the other modules are not necessary in this model. The temporal module however will play a more important role if we would construct a model of our experiment. For now, we will not model the experiment, but predict the outcome of the experiment from our models.

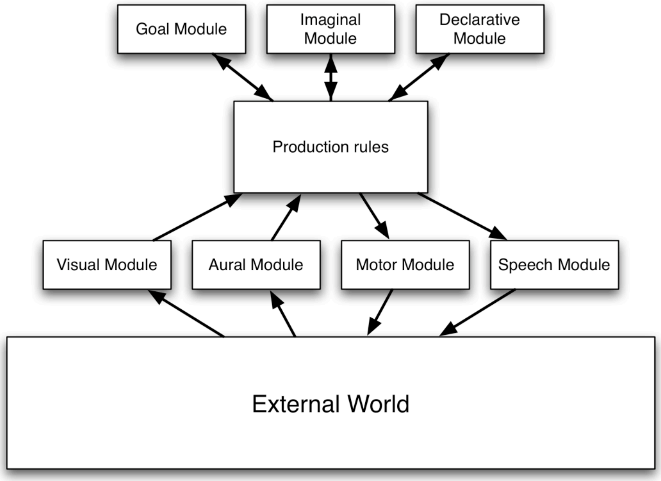

Figure 1.1. Modular layout of ACT-R. Boxes indicate information-processing modules, arrows denote information transfer.

1.3.2 Current models in ACT-R

ACT-R is a suitable cognitive architecture for modelling an LD task. With the available parameters we can tune the model to simulate the results of empirical data from LD experiments from the literature (Glanzer & Ehrenreich, 1979).

However, some adaptations of the LD task cannot be explained by ACT-R. For example a speeded lexical decision task (SLD), in which a signal tells the participants to respond faster than they normally would. When the decision has to be made before the necessary information is available, the participant has to 'guess' because the time interval needed to make a decision has been cut short by the signal (a deadline).
ACT-R can calculate the time needed for this decision, which we will call the 'Needed Decision Time' (NDT). After the NDT has passed, the information is available and a perfect score is reached.
This retrieval process cannot be further examined in ACT-R and has a ballistic nature (L Van Maanen & Van Rijn, 2007). From empirical data (Wagenmakers et al., 2004) however, it becomes evident that a deadline increasing towards the NDT gradually increases accuracy in participants' performance. If we were to cut short this NDT such as in SLD, ACT-R can simulate the outcomes by making use of noise. With an increasing deadline, the probability that noise facilitates the retrieval becomes higher. However, ACT-R cannot explain what happens *during* the retrieval process.

More importantly, ACT-R cannot explain results obtained in LD experiments in which some decisions for word stimuli take longer than the decision for a non-word. This is because the decision for a non-word in ACT-R is made based on a timeout. After this static amount of time has passed without enough evidence to support the decision for a word, the alternative non-word decision is made. Therefore, if some of the stimuli are so infrequently used in a language that they may require more decision time than the timeout, ACT-R cannot explain these lexical decision trials since it lacks the ability to do so. To be able to explain empirical data from literature (Wagenmakers et al., 2008) and simulate experiments where such infrequent stimuli are used, the method with which ACT-R makes the non-word decision has to be adapted.

1.4 The latency equation

The ACT-R latency equation is in principle not able to explain certain results observed in LD tasks, for example when retrieving different types of non-words, such as pseudo-words and real non-words. The retrieval of each non-word takes the same amount of time in ACT-R, while empirical data suggests otherwise by distinguishing non-words into pseudo-words and real non-words (Wagenmakers et al., 2004). ACT-R cannot simulate different latencies for non-word decisions, since the non-word decision is based on a timeout. When after a certain amount of time no evidence is found in favour of a word, the non-word decision is made.

The competitive latency equation (CLE) (Lebiere, 2001) is one of the proposed adaptations for the standard latency equation to overcome these problems (Van Rijn & Anderson, 2003). Competitive Latency means that the latency for a retrieval task is a function of the activation of all the other elements in the declarative memory. With CLE it is possible to simulate an (S)LD experiment with different types of non-words.

Currently in ACT-R, both with and without CLE, the NDT is determined at a fixed moment in time. The retrieval is then carried out and only after the NDT has passed a decision can be made. However, interference *during* the retrieval process can extend the NDT as seen in empirical data from picture-word interference (PWI) experiments (Glaser & Dungelhoff, 1984). Where the standard ACT-R latency equation is not able to explain these results, CLE also offers no consolation (Anderson, 2004). This is because both latency equations calculate the NDT from the activations of the chunks in memory and have a ballistic nature. Both cannot explain what happens during the retrieval process, and are therefore not able to explain results from experiments such as PWI.

1.5 Comparison of models for memory retrieval

The diffusion model (Ratcliff, 1978; Wagenmakers et al., 2008) relies on a decision mechanism that accumulates noisy information from a stimulus over time. How likely a stimulus is to be selected, determines the drift rate (arrow v in Figure 1.2). The drift rate indicates the average speed of accumulation towards the response boundaries a and b. In the case of an LD task, the drift rate is determined by how wordlike a stimulus is. For a frequently used stimulus (a frequently used word in the case of an LD experiment) the drift rate has a higher positive value than for a less frequently used stimulus, and a faster decision is made for response option A. For a non-word the drift rate has a negative value. In an LD task for example, the response option A would be the 'word' response and option B the 'non-word' response. A memory retrieval starts at point z in Figure 1.2, and once the dashed line (drift rate with noise on it) reaches one of the response boundaries a or b, a decision 'A' or 'B' is made.

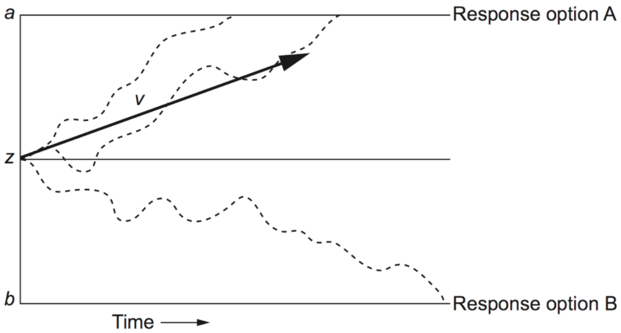

Figure 1.2. Diffusion model illustration, where the response time is the NDT to reach one of the response boundaries (Van Maanen, Van Rijn, & Taatgen, subm.).

The diffusion model is not capable of making decisions about choices *with more than two options,* as can be concluded from Figure 1.2, since there are only two response options available. The problems with this limitation can be further explained by theoretically extending the number of choices in a lexical decision task to the total number of words in the lexicon. Each word has an activation and can be obtained if its activation rises high enough, which is not possible in one retrieval process with the diffusion model. In an accumulator model (e.g. Vickers & Lee, 1998) this is possible, since in this type of model the increasing probability of a response option would not mean the decrease in probability for its alternatives.

Instead of a retrieval with only two possible outcomes, the result of a retrieval with a lot of elements, such as all the words in the lexicon, should be determined by including all elements in the competition for retrieval. This competition element shows more resemblance with the leaky, competing accumulator model (LCA) (Usher & McClelland, 2001) than with the diffusion model. Both of these models have not been integrated within ACT-R. The LCA model can handle more chunks in a competition to be selected and lets chunks influence each other, which is done by lateral inhibition. The LCA model uses a decay element, so that built-up activation does not last forever. However, the LCA model selects the chunk with the highest activation once the first chunk is past a static threshold. Problems with this method arise in theory when we would try to model an LD task with very low frequent words, where participants need more time for a ‘word’ decision then for a ‘non-word’ decision. With a static threshold and a time-out for the ‘non-word’ decision, modelling this would not be possible. This is because after the time has passed for a non-word decision to be made, no other decision can be made anymore. Therefore, the reaction time for the non-word decision is the slowest one possible in this type of model.

1.6 RACE

The LCA model was a starting point for the design of RACE, which stands for Retrieval by ACcumulating Evidence (RACE). RACE is embedded in ACT-R as an extension to the ACT-R declarative module, extending the possibilities for memory retrievals. RACE uses the same basic principles as the leaky competing accumulator model: a set of non-linear stochastic accumulators, each representing a chunk in the memory that can be retrieved. This means that the evidence accumulation for each chunk occurs in a non-linear way, calculated at each time step during the retrieval process.

Another key principle from RACE is that the activation of these chunks is decreased by decay, but increased by external input such as stimuli and by lateral excitation, rather than lateral inhibition as in the LCA model. By using excitation instead of inhibition, a chunk likely to be selected interacts with the chunks it has a strong relation with, instead of interacting with all the other chunks it has no relation with.

In RACE, a Luce ratio (Luce, 1963) applied to the activation of the chunks is used to determine the winning chunk. In a Luce ratio, the probability of selecting an item i from a pool of j items is given by the weight of that item, divided by summed weight of all the other items (Equation 1.1). This means, with a criterion of 0.95 as we will use, that the winning chunk automatically has a far greater activation than all the other chunks. The ratio is calculated each time step and when one of the chunks has a ratio higher than the criterion, this chunk is retrieved. The ratio and the criterion are explained in Equation 1.1, where the activation of a chunk relative to the activation of all other chunks in memory determines if the criterion is met.

$$P(i) = \frac{w_i}{\sum_j w_j} \Rightarrow \frac{Chunk\_activation}{Total\_chunk\_activation\_in\_memory} \geq Criterion \qquad (1.1)$$

RACE uses the activation at each time step to see if a chunk can be selected, so the time the retrieval will take is not known before the retrieval ends. Therefore a big difference with the CLE and ACT-R is that while in RACE the activation is used for selecting the chunk to retrieve, in the CLE and ACT-R the activation is solely used to calculate the time the retrieval will take, i.e. the latency. RACE is not bound by a static latency for the retrieval of a chunk from memory.

RACE leaves room for disturbances lengthening the retrieval process when retrieval has already started. This also means that an informed decision can be made with increasing accuracy as time passes (in e.g. an SLD task), in particular when the NDT has not yet passed.

1.7 Research question

In this project we will design a model that implements LD in ACT-R and in RACE. Next, we will model SLD based on RACE, which can implement the way participants deal with decision making when not enough information is available yet to make a decision. In this case we want to implement what participants decide when they do not have enough time to process whether they just read an existing word or a non-word. Reaction times can in theory be predicted by RACE, but not yet the proportion correct scores, which is needed to evaluate an SLD simulation. With our final RACE model and from the adaptations to RACE we will need to implement, we will predict the outcome of a combined timing and LD task.

We want to study the influence of timing on decision tasks, when this is implemented by performing a lexical decision experiment while focussing on a time estimation (TE) task. The manner in which timing influences the LD task will follow from our model.

The research question therefore will be the following:

"How can we design a model of memory retrieval tasks using RACE that can predict the influence of timing on the decision process when both tasks are performed simultaneously?"

1.8 Overview

We will construct the lexical decision task in ACT-R, simulate a lexical decision experiment from literature and match the empirical data. Next, we will show that RACE is able to generate the same results, to justify the backwards compatibility of RACE. Preliminary results show that RACE performs well qualitatively. Then we will extend the RACE model of LD beyond the capabilities of ACT-R and match more data from literature, to explain the need for the extra capabilities RACE has compared to ACT-R. We will show that RACE is also able to simulate tasks with missing information well in a qualitative manner, such as SLD. Finally we will conduct an experiment to see how time estimation influences performance on a combined timing and lexical decision task. The hypothesis we will try to verify is that when focussing on a time estimation task, performance in an LD task is worse when the LD stimulus is offered early in the time interval, than when the LD stimulus is offered later.

2 Model & Implementation

2.1 Lexical Decision in ACT-R

A lexical decision experiment consists of strings of letters that are presented to the participant, who then has to decide whether it is a word (W) or a non-word (NW). Such an experiment is done on a computer, and the participant has to press one of two possible keys. Experiments that we will focus on also manipulate word frequencies. High frequent words (HF) and low frequent (LF) words are used in combination with non-words. Participants have a lower response time for high frequent words than for low frequent words, while non-words take the most time from these three (e.g. Glanzer & Ehrenreich, 1979). We did not use different words per category, just one HF/LF/NW chunk to simulate the experiment. This is a simplification of which the implications will become clear in section 2.4, where we will justify this simplification.

2.1.1 ACT-R model of lexical decision

To design a model of an LD task in ACT-R, the 'subitize' model from the ACT-R 6.0 tutorial (unit 3) was used as a start-off point. This model displays a set of marks on screen and the participant had to count how many marks were presented. Unnecessary parts of the model were removed, such as the set of marks, and the ability to display an LD stimulus was added. The model now shows a predefined stimulus to the user. ACT-R 'reads' the stimulus into the visual buffer, simulating the participant reading the stimulus. This visual input is matched to a text chunk if the word is known, which means that the grammatical form of the word is recognized for existing words. If the stimulus is a non-word, the non-word text chunk will be retrieved.
The model does not yet have the ability to respond W or NW after retrieving the text chunk, since both chunk types are the same. The difference lies in the spreading activation from text chunk to lemma chunk. A lemma is an abstract form of a word in the mind (Levelt, 1989). Therefore, spreading activation from text chunks to lemma chunks can only occur for existing words.

When the stimulus is not perceived (simulating for example distracting the participant so that the stimulus is missed) no chunk can be found and the threshold will be returned instead of a text chunk. This signifies a mistrial and will be excluded from the results. After returning a valid text chunk, the meaning of the text in the text chunk is retrieved from memory in the form of a lemma chunk. When a lemma is found, the answer is given by virtually pressing the key for the 'word' decision on the keyboard through the motor module. When a valid text chunk was found but no matching lemma could be found, as is the case for the non-word text chunk, the key for 'non-word' is pressed.

2.1.2 Empirical data

When the stimulus is presented to the participant there are three categories of possible reaction times (RT's), one for each type of stimulus. These categories and RT's come from empirical data (Glanzer & Ehrenreich, 1979). From the data it is clear that the RT's are ordered in the following order of time it takes to retrieve the chunk: HF < LF < NW. Values obtained from the data from Glanzer are from mixed lists. In these lists high, medium and low frequency words are mixed with non-words. HF is here defined as occurring more than 148 times per million words, medium frequent as 6 to 8 occurrences per million and LF as less than 2 per million. We are only interested in the HF, LF and NW RT's, since recent literature mostly mentions these categories. The mean RT's are 536ms for HF, 678ms for LF and 757ms for NW.

2.1.3 Model settings

To achieve the aforementioned RT's in our model the parameters for the retrieval threshold (rt), the F factor (lf), which is used for scaling, and the base levels of activation of the word chunks may be modified. Retrieval time in ACT-R is determined by the activation of each chunk; the higher the activation, the faster the retrieval. The scaling factor is a global parameter and scales all retrieval times with the same factor.
First the retrieval threshold was set at 1.15 to get the RT for the HF word at 536ms. For this, the base-level activation of the HF chunk was set sufficiently high above the rt, at 3. Next, the lf factor was set to 0.83 to scale the RT of the NW to its desired value. And finally, the base level activation of the LF chunk was set at 1.51 so the desired RT was reached for this chunk. With these settings, the RT's of Glanzer were exactly matched (see Table 2.1 in section 2.2.3). For purposes of comparison: the non-words in the experiment from Glanzer are best classified as lexical non-words, since they are pronounceable.
From these results we can conclude that ACT-R is capable of explaining results in a simple lexical decision task.

2.2 Lexical Decision in RACE

2.2.1 Theory of RACE

RACE is a new model for retrieval from declarative memory (L Van Maanen & Van Rijn, 2007) in ACT-R and is based on competition between the chunks in the declarative memory. The decision of which chunk to retrieve is not made solely based on the highest activation among the chunks, but on the Luce ratio of each chunk. This ratio provides a factor between 0 and 1 of the relative activation of that chunk, with respect to the sum of all activations, i.e. the total activation in the memory. Therefore, if one chunk has a big part of the total activation in the memory, this chunk is selected. On the other hand, if more chunks have a high activation but do not differ from each other very much in activation, the process will not decide yet. This is different from other models with a static threshold, such as the LCA model, where the decision is always made at the latest at a timeout.
The activation of each chunk in ACT-R is solely used to calculate the NDT. When the NDT has not been reached yet and a retrieval is made, in ACT-R there is no information available about which chunk is more likely to be retrieved than others (Figure 2.1, left).

In RACE however, evidence accumulates between onset and retrieval. Therefore, if the retrieval interval is cut short, RACE can calculate the activation for each chunk at a specific time step. Now a comparison of activation between chunks can be made, and RACE has the ability to make an informed decision about which chunk to select (right).

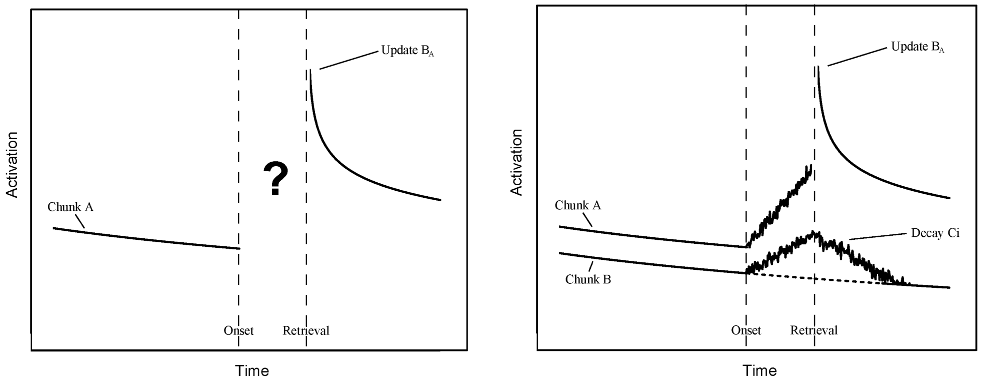

Figure 2.1. Left: ACT-R retrieval process with no information between onset and retrieval. Right: RACE retrieval process with accumulating evidence between onset and retrieval.

2.2.2 Differences ACT-R and RACE

Apart from its aforementioned ballistic nature, ACT-R poses more problems when trying to model LD experiments. For example, the non-word decision is always made within a certain (again, static) time interval. From recent experimental data (Wagenmakers et al., 2008) it becomes clear that wrong decisions regarding HF and LF (and even very low frequent, VLF) words take up different amounts of time. A wrong non-word decision when the stimulus is HF for example, when participants focus on accuracy instead of speed, takes less time than a wrong LF non-word decision. This means that the non-word decision cannot be based on a timeout, since that can only generate a single RT for a non-word decision, but has to work in another way. Modelling this in ACT-R is currently not possible. Since RACE does not use a static timeout it is able to simulate such results.

A practical difference is that RACE uses discrete time steps, which can be set to a specific value, to generate output. In the following sections we will see that ACT-R results can be tuned to the millisecond since this is a continuous approach of the process. ACT-R uses a more abstract algebraic model of the retrieval process than RACE, which is in principle independent of the time in ACT-R. RACE is considered a process model, which relies on sequential sampling. Since 5ms is a number that is applicable to more processes in the brain, such as firing rates of neurons (Coon, 1989), RACE results are often multiplications of 5ms by setting the frequency parameter in RACE to 200Hz.

2.2.3 Matching ACT-R and RACE results in lexical decision

With the working LD model in ACT-R as a basis, RACE was used in combination with ACT-R to generate the results. The goal is to make RACE generate the same results as ACT-R, without changing parameters in the ACT-R part of the model. If we were to change those parameters, the outcome of the ACT-R model would change again. So without changing this model we want as much flexibility in our choice of parameters for the RACE part as possible. Therefore a series of good fits was determined in ACT-R instead of just a single fit. With these series a simple model was made of the ACT-R results. This was done with an Excel model (extrapolation) of the ACT-R model and the connection between its parameters and the outcome (RT's). As a result the base-level activation of either the HF or the LF word may be set arbitrarily, and from that the rest of the parameter values for the ACT-R model follow so that this fits the data again. This gives us the flexibility of changing the base-level activation of either the HF or the LF word in RACE to a suitable value.

With this model giving us the flexibility we needed, a suitable set of RACE parameters was determined. Since there are too much parameters in RACE to use trial-and-error with random parameter settings, the influence of each parameter was determined. While keeping the other parameters at set values, each parameter was in turn changed to determine the change in results. In this way, interaction results are omitted. We chose to omit these because we expected that interaction terms were not needed to obtain good results. Also, we did not think interaction terms would cause problems in obtaining good results.

When the influence of the parameters was determined, the parameters were adjusted to fit the model on the experimental data. For some of the parameters smart values were chosen based on reasoning about those parameters. Other parameters that did not have such constraints were set according to the influence they had on the results.

With these final parameters, the model produces the same results as found in the experimental data, both with and without RACE (Table 2.1). This result suggests that RACE can model an LD experiment with the same outcome as ACT-R, as we claimed earlier.

Condition	Empirical data	ACT-R model	RACE model
HF	536	536	535
LF	678	678	680
NW	757	757	760

Table 2.1. Comparison of empirical RT data in lexical decision with both our ACT-R and our RACE model.

2.2.4 Noise addition and distribution modelling

Next, we want the model to be able to produce RT distributions as well, which means performing a great number of trials, where the use of noise makes the RT variable over all trials. Without noise, the RT is fixed as can be seen in the previous section. To achieve these distribution results, we will extend the RACE model to include the noise parameter from ACT-R (:ans). Each retrieval can now be speeded up by noise adding activation to a chunk, or slowed down by noise subtracting activation from a chunk.

The median correct reaction times were determined by performing a large number of trials. The amount of noise has influence on these medians, but also on the variance, i.e. the width of the RT distribution . The distribution was fitted so that the shape (right-skewed) and the median correspond to the empirical data (Wagenmakers et al., 2008) as we show in section 2.4. An example of a memory retrieval with noise is shown in the trace in Figure 2.2. Here the LF word is retrieved.

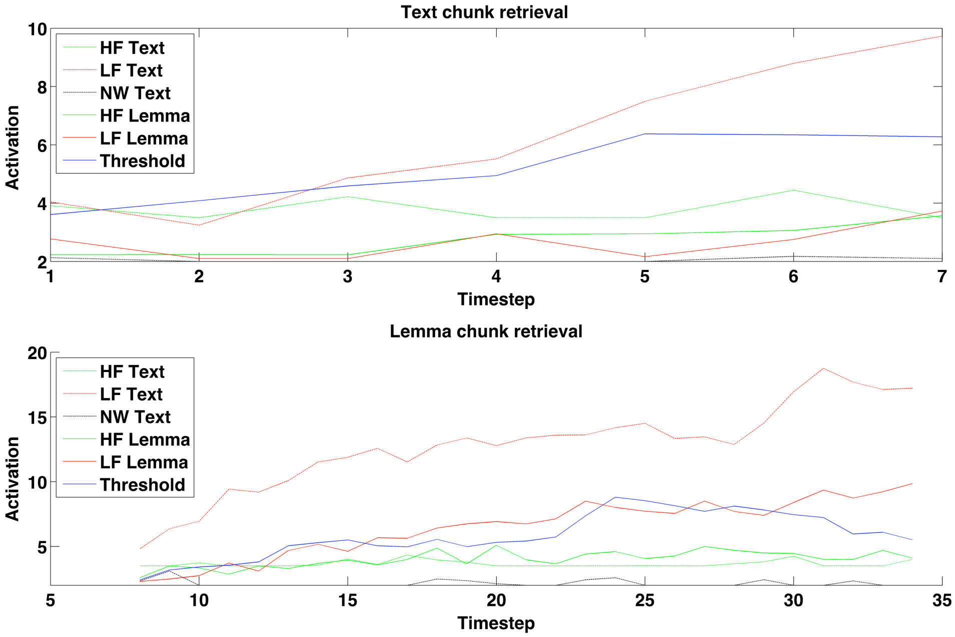

Figure 2.2. Retrieval of an LF chunk from memory. Above is the text chunk retrieval, below the lemma chunk retrieval.

In the upper graph we can see the retrieval of the LF text chunk from memory, with noise. The competition is only between the text chunks (dashed lines) and the threshold. If the stimulus is an LF or HF word, the corresponding text chunks should be retrieved. The same goes for the non-word. If the individual cannot read the letters (for example, the screen is blurred) then the threshold should be retrieved, which signifies a mistrial. In the lower graph the lemma chunk is retrieved, in a competition between all lemma chunks and the threshold. If the word does not exist in the lexicon of the individual, the threshold is retrieved, signalling a non-word. Although the text chunks do not compete anymore in the lemma retrieval, they still influence the outcome by spreading activation to their corresponding lemmas. The LF text chunk continues to rise as well, because the stimulus is still present on the computer screen.

A production rule fires in between the retrieval of the text chunk (end of the upper graph) and the start of the retrieval of the lemma chunk (start of the lower graph). This production rule has the condition that a text chunk is retrieved, and starts the retrieval of the lemma chunk. This process takes time since the fact that letters were recognized has to be passed on to the module where the lemma information is stored; therefore the activation of all chunks decays during this period. So although the time steps continue

from 7 in the upper graph to 8 in the lower graph, there is an interval without RACE activity in between.
The retrieval is finished when the Luce ratio of one of the chunks is far higher than that of the rest and reaches the criterion, which is always set at 0.95 in our research. The Luce ratio of the LF text and lemma chunks can be seen in Figure 2.3. In both the text chunk and the lemma chunk retrieval, the blue line indicating the Luce ratio reaches its criterion in the last time step displayed, i.e. time step 7 for the text chunk and time step 34 for the lemma chunk.

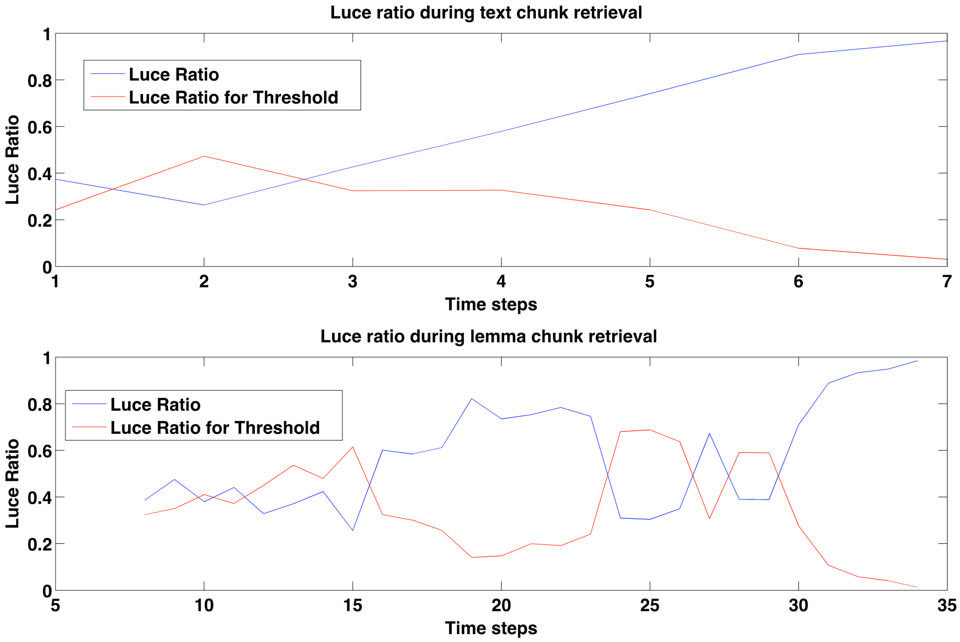

Figure 2.3. Luce ratios during retrieval of text chunk (above) and lemma chunk (below).

At the start of the LF text chunk retrieval there are more chunks with such a high Luce ratio that they can still be retrieved (not shown in Figure 2.3). As the retrieval process continues, the Luce ratios for the other chunks go to zero since their activation becomes very low compared to the activation of the LF chunk. At the end of the retrieval, the sum of both Luce ratios (LF and threshold) shown in Figure 2.3 becomes nearly one. This means that no other Luce ratios play an important role anymore at that point in the retrieval process.

2.3 *Speeded lexical decision in RACE*

The LD task can be made more challenging by setting a deadline for the response. This is called speeded lexical decision (SLD). In some cases the participant doesn't have enough time to completely process the string (Figure 2.4), and therefore has to guess whether the string represented a word. In this type of experiment the proportion of correct answers is measured instead of the reaction time.

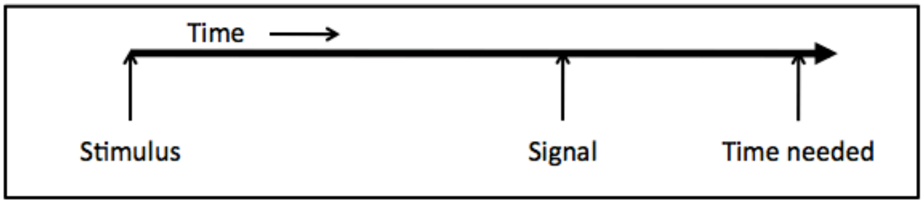

Figure 2.4. Acting on a stimulus when a signal is received instead of when the participant is ready.

The manner in which the SLD procedure differs from the LD process can be illustrated from the literature. An experiment was carried out (Wagenmakers et al., 2004) to verify a Bayesian model of memory retrieval (Zeelenberg, Wagenmakers, & Shiffrin, 2004), in which people receive a stimulus and 2 tones are played. At the 1st tone a letter string is presented, which is followed by a 2nd tone. Next, the lexical decision has to be made before or at the 3rd evenly spaced (imaginary) tone. People tend to get rather annoyed with this experiment, *feeling* they have to guess all of the time. This is because the participants' memory retrieval process for the letter string is cut short by a signal that tells the participant to respond. The results show that with an increasing deadline participants perform increasingly better than chance levels. These results therefore provide evidence that the static latency approach is not the best way to model memory retrieval.

With adaptations to RACE it is possible to simulate an SLD task by passing the deadline at which the retrieval has to be made. The deadline in ms is passed to RACE, and for the sake of the model the time that everything **but** the RACE retrieval takes is known. This is called the non-decision time (Wagenmakers et al., 2008), although in RACE this non-decision time is the same in each trial since it only depends on the execution of production rules. We subtract this time from the deadline and know how much time RACE has to decide. RACE now checks every cycle if this time has passed and if so, returns the chunk with the highest activation (with a certain probability). No chunk has to reach the Luce ratio criterion, and in this case the chunk with the highest activation also has the highest Luce ratio. As a result, other chunks with almost the same activation as the winner chunk do not delay the retrieval in the speeded condition of the experiment, since the decision has to be made at a certain time step. So if two activations are nearly the same at the last time step before retrieval, the noise over the last time step determines which chunk gets retrieved. The earlier in the interval, the smaller the difference in activation between the chunks. This implements the empirical result (e.g. Wagenmakers et al., 2004) that more mistakes are made when less time is available.

The model was extended with this kind of functionality and a qualitative fit was generated (Figure 2.5) for deadlines of 75, 200, 250, 300, 350 and 1000ms. The 200ms data points for LF and HF data are lower than they should be, which is due to model settings. The rest of the data points show a reasonable fit, although the start is still slightly too high (around 60%). With enough time (here 1000ms), all three conditions approach a perfect score as is the case in reality with such tasks. The model was not tuned to generate results that can verify empirical data, but was just adapted to add the functionality of signal-to-respond tasks such as SLD. Since SLD is not the focus of this research, we will not explore this type of task further.

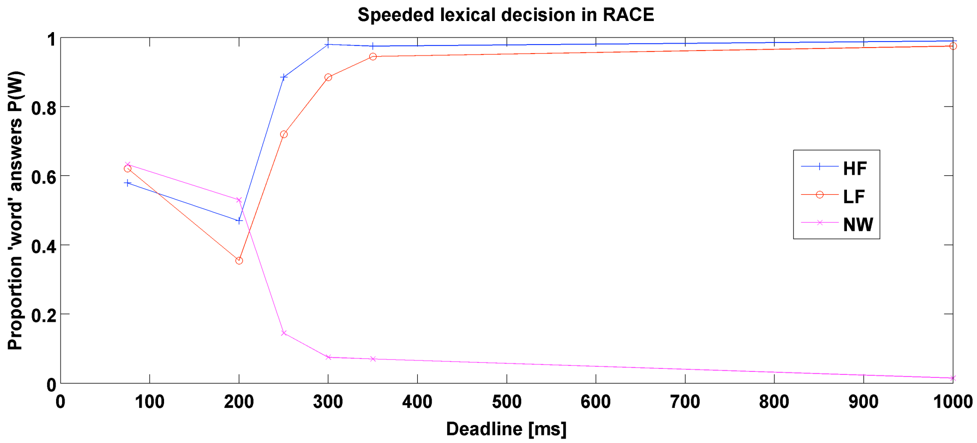

Figure 2.5. SLD simulation, qualitative fit. For an increasing deadline, fewer mistakes are made.

In future SLD experiments, it will be a matter of tuning the model, which is already capable of generating results for SLD type experiments.

2.4 VLF condition: Extending the RACE model

Now that we constructed the same simplified model in ACT-R and RACE, we no longer took into account the limitations of the ACT-R model. From here on, the RACE model was extended beyond the capabilities of ACT-R. To be able to model the results of recent LD experiments (Wagenmakers et al., 2008) we added a category of very low frequent (VLF) words. What's very interesting in the outcome of this experiment, is that the RT's are ordered as HF < LF < NW < VLF. In other words, the lexical decision for a VLF stimulus takes more time than an LD for a non-word.

Next to varying the word frequency, the instructions to the participant also were manipulated in this experiment. In the 'focus on accuracy' condition, participants were told to respond as accurately as possible, where in the 'focus on speed' condition the instruction was given to respond as quickly as possible. We modelled the data from the 'focus on accuracy' condition, since the effects of word frequency manipulation are more pronounced in this condition.

The diffusion model, although it has the limitation of generating only two possible answers as the outcome of a retrieval, can model these results well (Wagenmakers et al., 2008). There can be more types of words, so in theory more choices to decide from, but the answer in an LD task is always W or NW. Therefore, since there are only two possible answers the diffusion model is not hindered by this limitation in this task.

In the diffusion model, as opposed to deadline models with a temporal timeout mechanism for non-word responses, the non-word responses are generated with the same decision mechanism as the word responses. This makes it possible for a response on a VLF stimulus to take more time than the response on an NW stimulus in the diffusion model.

For our model to be able to deal with a situation in which some word decisions take more time than NW decisions, the way in which the threshold behaves had to be changed. In ACT-R and the basis of RACE the threshold value is static and is used as a timeout. Therefore, it is not possible to have an RT for VLF stimuli that is higher than the RT of the non-word, so we modified the threshold to an increasing threshold without timeout. Because of the increase with time, the threshold behaves as a chunk and can now reach a Luce-ratio of 0.95 as well and be selected.
Increasing the threshold in other ways than with the same accumulating activation function as the text and lemma chunks results in unwanted behaviour. For example with a quadratic increase, it's not possible to attain RT's close to but under the threshold RT. Since the VLF chunk is retrieved slower than the NW (threshold) chunk, this way of threshold increase is not suitable.
The value with which the threshold increases is a parameter and can be manipulated. By increasing the threshold in the same way as the text and lemma chunks, the RT can grow very large. This happens because the Luce ratio criterion is reached very slowly when the threshold and a text or lemma chunk have almost similar increases in activity. This is necessary for modelling the VLF response, which is shown in Figure 2.6. For illustration purposes the noise has been disabled here, so that the difference between threshold and VLF lemma chunk is clearly visible.

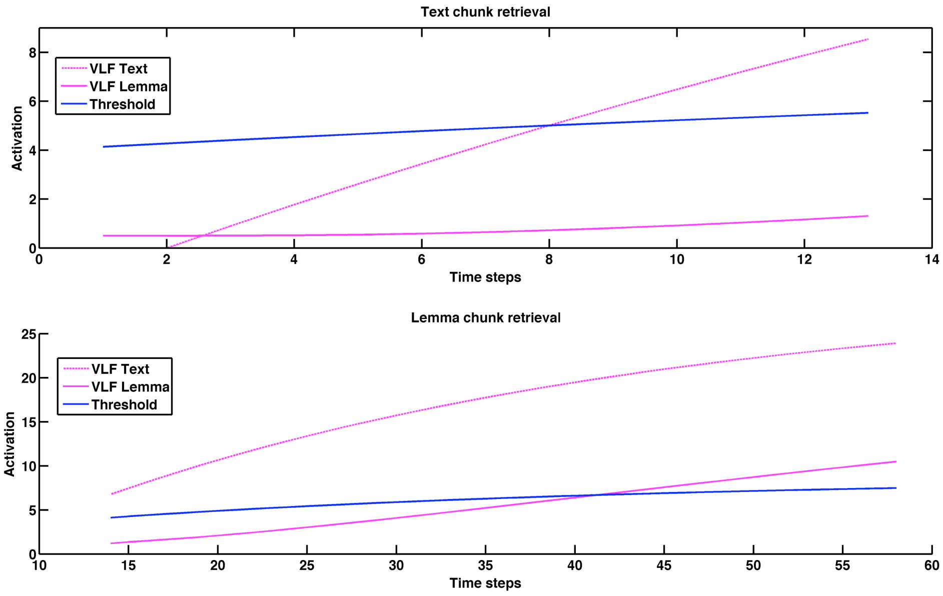

Figure 2.6. Retrieval of a VLF chunk from memory. Above is the text chunk retrieval, below the lemma chunk retrieval.

In the upper graph, the VLF text chunk is retrieved after some time. Next the lemma retrieval starts in which the threshold and the VLF lemma chunk increase both quite slowly. After some time, the VLF lemma chunk is retrieved. The activation of the VLF lemma chunk continues to rise as a result from the spreading activation from the VLF text chunk, indicating the VLF stimulus is still visible on screen in the LD task.

We can compare this retrieval with a (faster) NW retrieval in Figure 2.7. During the text chunk retrieval the NW text chunk rises faster than the other chunks and is retrieved. In the lemma retrieval the NW text chunk is no longer displayed, since it spreads activation to no other chunk and therefore has no influence anymore. Instead, the threshold is retrieved during lemma retrieval, which indicates an NW decision.

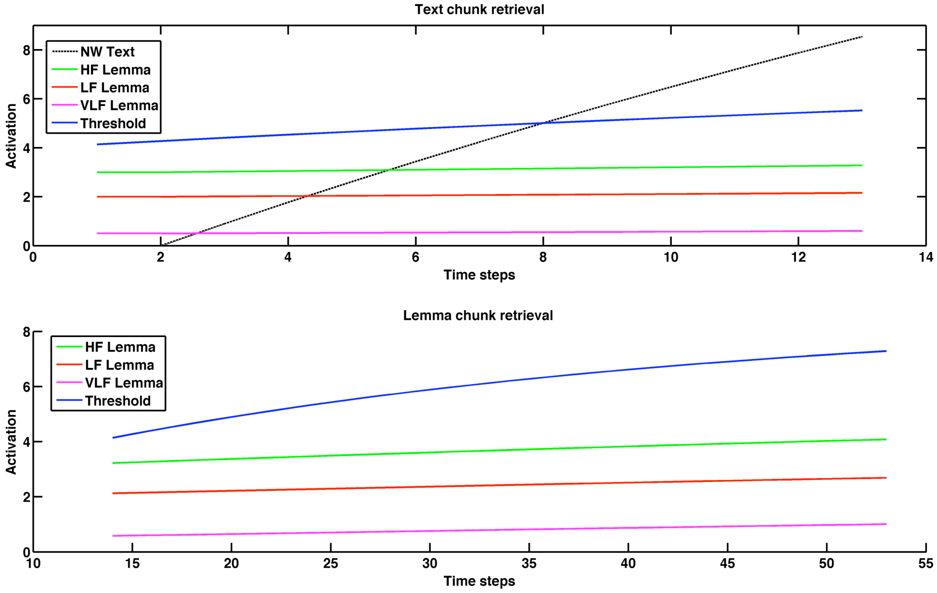

Figure 2.7. Retrieval of an NW chunk from memory. Above is the text chunk retrieval, below the lemma chunk retrieval.

The noise that is used for the text and lemma chunks has been added to the threshold as well, to create the same behaviour for the threshold as for the other chunks. Since the noise is relatively large, one can understand with Figure 2.6 in mind that noise will have a large influence on whether the VLF chunk is retrieved instead of the threshold and in which time span. Therefore, more errors will be made in the VLF chunk retrieval process than for the other chunks, which corresponds to empirical data (Wagenmakers et al., 2008).

With the added VLF condition we modelled this empirical data, which were the median values as well as the shape of the distributions. Since reaction time data is generally right skewed (Mccormack & Wright, 1964) we modelled this by making use of noise, as can be seen in Figure 2.8. The earlier in the retrieval process, the more influence noise can have. Since all chunk activations are relatively low at the start of the retrieval, the noise value can cause a higher proportional increase in chunk activation. Later in the retrieval process, all chunk activations are higher and the noise addition has a smaller impact on the total activation in memory. This relatively large influence of noise at the start of the retrieval causes a high proportion of all RT's to be concentrated closely together. The longer the retrieval process is underway, the less influence noise has and the more spread out the RT's become. In other words, variance in reaction times increases with time. This results in a right skewed distribution of reaction times.

A comparison of the results of our model with empirical data can be seen in Figure 2.9, where it is still clear that our model produces similar shaped (right skewed) RT distributions. The five plus signs for each condition indicate the 0.1 / 0.3 / 0.5 / 0.7 / 0.9 percentiles, the median plus sign is in bold. In our model the variance in each condition is less than in the empirical data, which can be ascribed to the representation of the frequency conditions in our model.

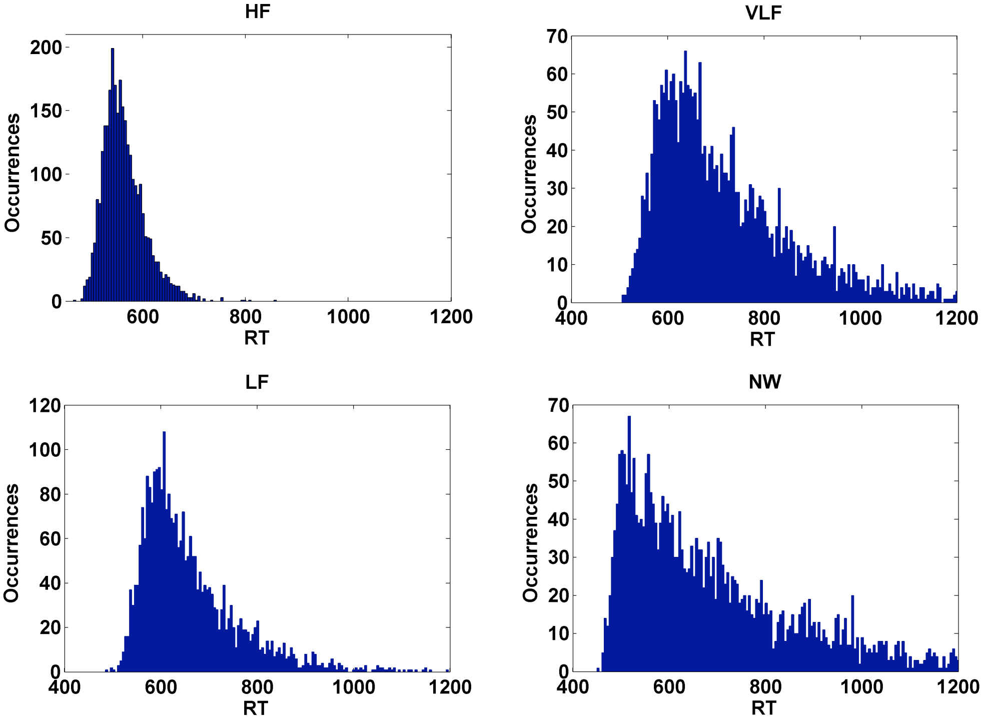

Figure 2.8. Right skewed distribution of RT for all conditions in our RACE model, focus on accuracy. All horizontal axes are cut off at 1200ms, therefore not all data is visible.

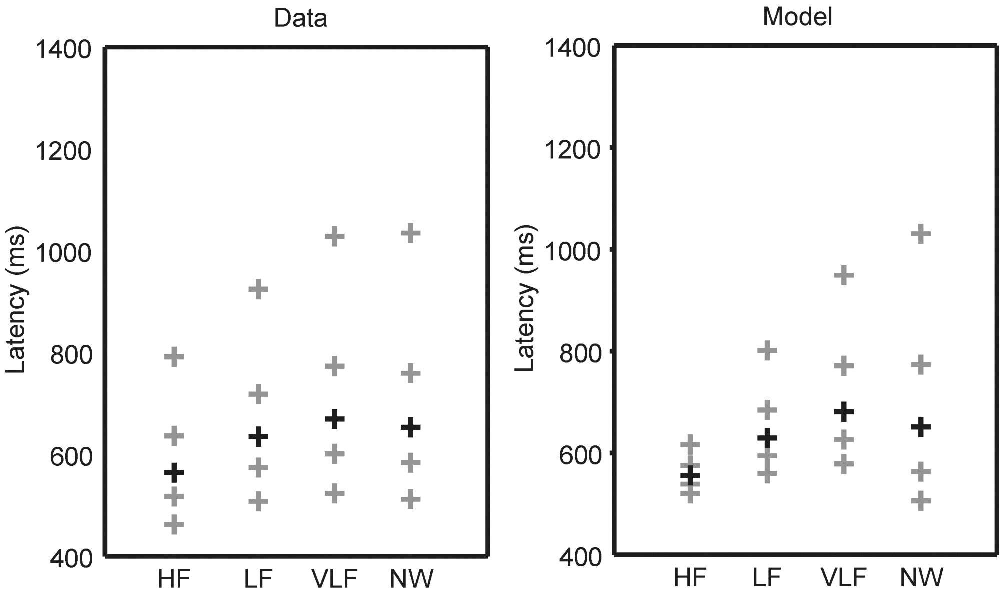

Figure 2.9. Modelling RT median values and distribution shapes, compared to empirical data.

When we look at the empirical data in Table 2.2 (Wagenmakers et al., 2008), we see that wrong decisions in 'focus on accuracy' conditions take about the same amount of time (HF condition) or more time than the right decisions. This is probably because with HF and (V)LF words, we scan through our known words and decide 'non-word' only when no words were found in the search. When we focus on accuracy, we make sure that the stimulus does not have a lemma associated with it and only then answer non-word. Therefore, finding a word in this search results in a faster response.

In the 'focus on speed' condition, we clearly see that a lot of wrong decisions are the 'too quick' decisions; the error RT's are all smaller than the correct RT's. In this condition, participants respond too quickly and therefore make the wrong choice, i.e. 'fast errors' (e.g. Link & Heath, 1975).

Stimulus	Focus on accuracy	Focus on speed
HF Correct RT	564	471
HF Error RT	563	441
LF Correct RT	636	510
LF Error RT	653	480
VLF Correct RT	674	525
VLF Error RT	760	498
NW Correct RT	655	508
NW Error RT	718	488

Table 2.2. Empirical data, median reaction times (in ms) for different conditions (Wagenmakers et al., 2008).

From this data we used the medians of the correct RT for the 'focus on accuracy' condition. The comparison with our model is shown in Table 2.3.

Condition	Observed median correct RT	RACE Model median correct RT
HF	564	555
LF	636	630
VLF	674	680
NW	655	650

Table 2.3. Observed correct RT from accuracy condition (Wagenmakers et al., 2008) vs. the results generated by our RACE model.

The table reveals that the maximum deviation from the model is 9ms (HF condition), which would mean 2 discrete time steps measured in RACE values. The root mean squared deviation from our model with the data is 6.7ms. Since it is already clear from Figure 2.9 that the model does not have the same kurtosis (our model is more ' peaked' since the variance is smaller) we did not compare the kurtosis values with the data, or the skewness values. This will be useful when the variance is bigger in our model, as we discuss in the last chapter.
To summarize, given the shape of the distributions and the median RT values RACE already models the empirical data quite well. Since modelling the VLF RT requires an increasing threshold and the increase of the threshold occurs with time, time is a critical aspect of our model and in our experiment which follows from the model.

2.5 *Non-linear timing model*

The Cognitive Modelling department here at AI in Groningen have developed a theory to implement time perception into ACT-R (Taatgen, van Rijn, & Anderson, 2007), based on the pacemaker-accumulator internal clock model (Matell & Meck, 2000). In this type of model (Figure 2.10) an accumulator counts the steady stream of pulses that is produced by an internal pacemaker. The start of the count is signalled by the opening of a switch, and the accumulated value of pulses is stored in memory after the end of the interval. When the interval has to be reproduced, a new interval starts and the number of elapsed pulses is constantly compared with the value stored in memory, until both values are equal.

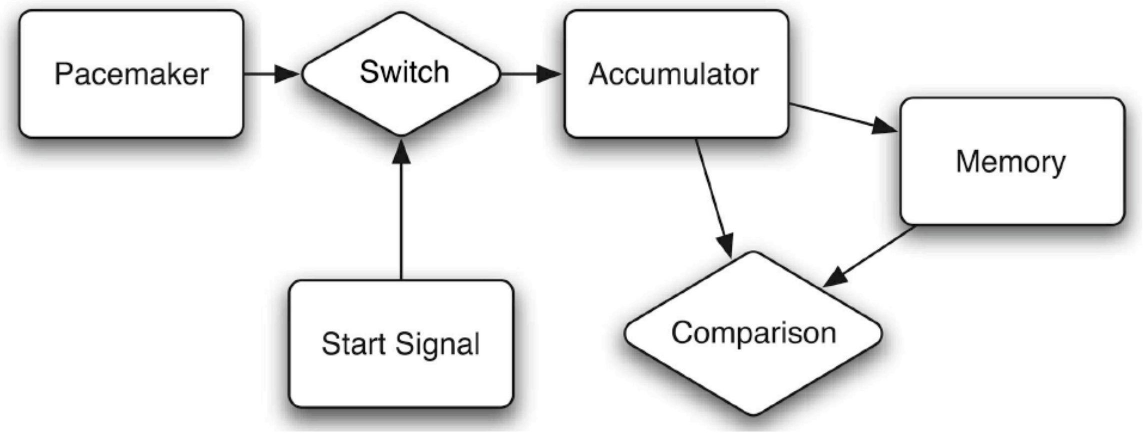

Figure 2.10. Pacemaker-accumulator internal clock model (Taatgen et al., 2007).

We agree with Van Rijn & Taatgen (Van Rijn & Taatgen, 2008) that the pacemaker in this configuration does not produce pulses with a constant interpulse interval, but produces pulses that are spaced apart with increasing intervals. In their research, they state that if time perception is linear, it is possible to perform simple temporal arithmetic on their results without systematic biases. However, the results of their experiments show that time intervals may not be linearly added or subtracted, which supports the hypothesis that time perception is non-linear. We will make use of the non-linearity of time perception in our experiment.

To achieve this non-linear time perception, the start pulse is set to a fixed value. The interval leading to each subsequent pulse is defined as *a* times the interval between the previous two pulses, after which noise is added. The spacing of the non-linear perception of time is presented in ticks, as can be seen in Figure 2.11, with respect to the linear time in seconds.

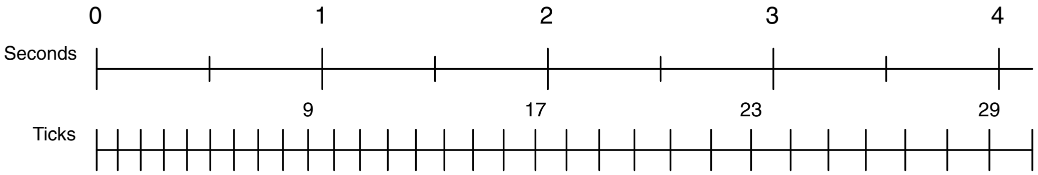

Figure 2.11: Non-linearity of time perception in ticks (Van Rijn & Taatgen, 2008).

For our experiment the non-linearity of time perception is vital, since the time perception task influences performance on the LD task as will be explained in the next chapter.

3 Experiments & results

3.1 Experiment

The experiment was designed to test the hypothesis that since time perception is non-linear, a time estimation task influences the performance on a lexical decision task. We tested this hypothesis in a simultaneous lexical decision and time estimation experiment. The time from the beginning of the time interval to presentation of the LD stimulus is called stimulus onset asynchrony (SOA).
The influence of non-linear time on an LD task becomes clear when we consider the evidence accumulation for the different chunks. The accumulation for HF/LF/VLF chunks is continuous during the time interval, since it is generated by the stimulus that is constantly on the computer screen. Due to spreading activation and decay the representation of this continuous accumulation is not a straight line in Figure 3.1. The threshold, however, accumulates with the internal time since the threshold gains activation by the lack of evidence for word conditions. This means that with every tick the threshold value is increased. Therefore the threshold accumulates relatively fast in the beginning of the interval when the ticks are spaced more closely together. We believe that this results in more wrong 'non-word' choices at low SOA, since a relatively faster increasing threshold means more non-word decisions. The effect is stronger for LF than for HF stimuli, since the HF chunks accumulate faster.

The theory for this hypothesis comes from the non-linear character of time perception (Van Rijn & Taatgen, 2008). We assume that internal time steps, 'ticks', in the individual's perception have more space in milliseconds between them as the interval increases. Therefore, when measured in milliseconds the time steps follow each other more closely in the beginning of the interval than later in this interval.

From the illustration of the main effects (Figure 3.1) it becomes clear how the non-linear time steps are spaced, as indicated by the black line indicating the internal time representation. Accumulation of a word chunk and the threshold can be seen at the 250ms SOA and the 1750ms SOA. The lines of the word chunks have the same shape as the threshold, since the threshold accumulates in the same way as word chunks. Noise has been left out of this illustration for obvious reasons of clarity.
The increase in activation for the threshold is not linear, and so is the spacing between the ticks. Therefore, if we consider the increase in activation per tick this approaches a more linear value for threshold increase, simplifying the model.

We predicted that the time estimation task would degrade the performance in LD trials where a word stimulus was presented at low SOA compared to trials with a high SOA, with a stronger decrease in performance for LF than for HF stimuli. Also, RT's would go up for word stimuli. In LD trials where a non-word stimulus was presented, we predicted that the accuracy would increase and the RT would decrease at low SOA.

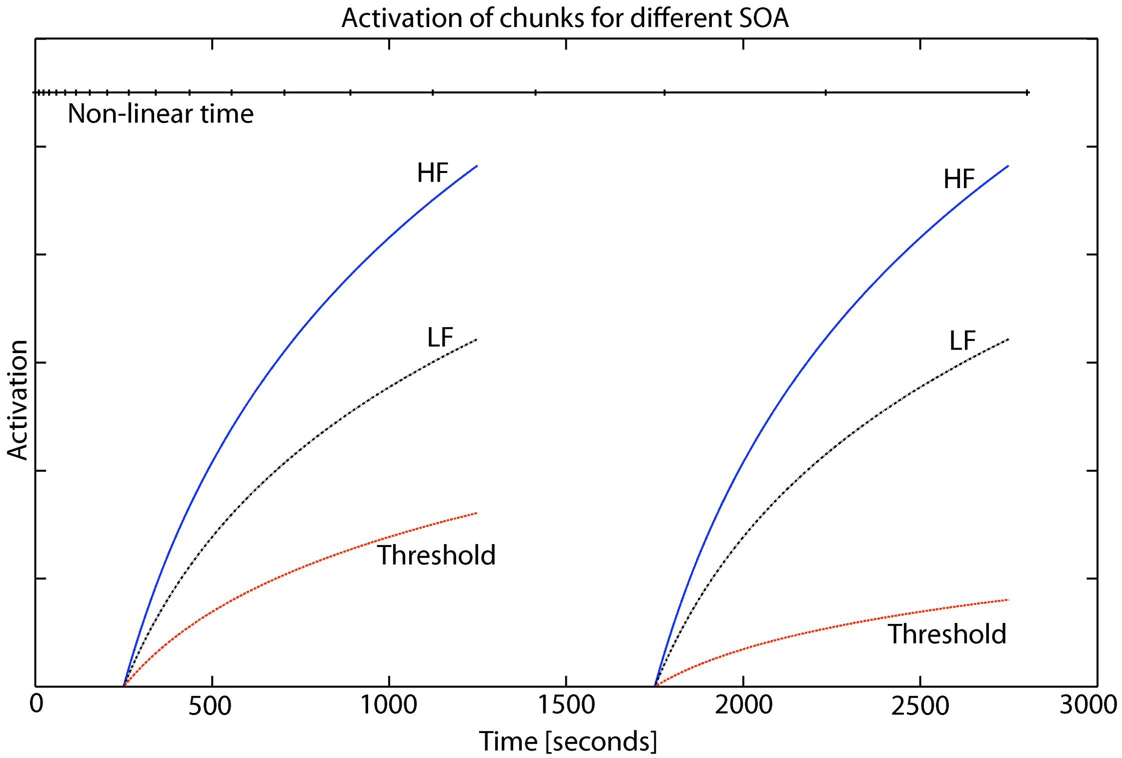

Figure 3.1. Illustration of the main effects. For lower SOA, the threshold rises more quickly since the ticks which control time perception are spaced more closely together. This results in more 'non-word' answers at low SOA than at high SOA. Since the relative difference between threshold and the LF chunk is bigger for different SOA than for the HF chunk and the threshold, the effect is stronger for LF stimuli than for HF stimuli.

3.2 *Method*

3.2.1 Participants

35 participants took part in the experiment, all were native German speakers.

3.2.2 Materials

We used a set of German words extracted from a CELEX database, which was checked by a native German speaking researcher here at the AI department. She made sure that strange words were removed, for example names and words with a funny connotation. Only normal German words were kept. All words were between 3 and 6 letters in length. Out of these words, 88 were high frequent words (occurrence in German: more than 10 per million), 88 were low frequent words (occurrence in German: 2 or 3 per million) and 176 were non-words. These non-words were 'Dutch non-words', and were therefore checked against the 'van Dale online German dictionary' provided by the University of Groningen, to make sure that these words also had no meaning in German. Three words that did were replaced with checked others. All words and non-words were written with a capital, as is the rule in German for nouns, to create a uniform set of stimuli. Otherwise, the use of a capital might have been an indication that a real word was presented. All conditions were matched for word length.

3.2.3 Design

The aim of the experiment was to study the influence of the SOA on the performance in an LD task. The critical manipulation was to offer the LD stimulus at 3 different SOA's during the time estimation task, at 250 / 1000 / 1750 ms.
Points are awarded if the participant makes the right choice, motivating the participant. A right guess for the time interval results in 100 points, a wrong guess yields no points. For the LD task, a right decision yields 25 points, whereas a wrong decision gets the participant -25 points. These values were chosen as a motivation to concentrate mostly on the time interval, so that the internal time estimation task could have a big influence on the performance in the LD task. The subtraction of points for a wrong answer in the LD task was added to make sure that the participant did not neglect the LD task.

3.2.4 Procedure

For the experiment we used a computer with Windows XP and E-prime software (Schneider, Eschman, & Zuccolotto, 2002) to execute the experiment. When describing the experiment below, it should be noted that during the experiment all instructions and feedback was given in German.

First the time estimation task was trained. After the instruction the participant saw a white cross on screen for 750ms, which signalled the participant to prepare for the task. This screen was followed by a screen with a purple circle, which was shown indefinitely. After an - to the participant unknown – interval of 4 seconds, the participant should press the spacebar. The margin lay between 3750ms and 4250ms. Each time the participant pressed the spacebar, feedback was shown. Feedback could be 'Right', 'Too Slow' or 'Too Fast'. In this fashion, the participant could learn the interval without any prior knowledge of the length. After 30 trials, the participant was assumed to have learned the interval.

Second, the lexical decision task was trained in 20 trials. After instruction, the participant saw the same white cross as before for 750ms in preparation before the stimulus was shown. The LD stimulus was displayed for 750ms, after which the standard feedback 'Too slow' was given if no answer was received. When the participant pressed '1' or '2' while the stimulus was on screen, either 'Right' or 'Wrong' was given as feedback. For all participants with an odd subject number, '1' was assigned to the word choice and '2' was assigned to the non-word choice. For the participants with the even subject numbers, this was vice versa. The instructions stated for each participant which keys to press.

Third, the instruction was presented that both tasks would be combined. The participant practiced 27 trials with the combined task. In this task and the rest of the experiment, the

participant had 1000ms to handle the LD stimulus, instead of 750ms as during the single task LD training trials. This was decided after a pilot study with the first 11 participants, which showed some problems with responding within 750ms in the combined task. Also, we wanted to ensure that the participant did not expect an LD task each trial. Therefore the LD stimulus was present in 20 trials, whereas in the other 7 only a time estimation task had to be performed.
When the practice sessions were finished, the participants' score was reset and the instruction appeared that the experiment would start.

The experiment itself started with 16 filler trials, followed by 400 experiment trials. Of these 400 trials, 100 contained only the time estimation task. For the remaining 300, 150 were non-words, 75 were HF words and the last 75 were LF words. After 200 trials the participant was offered the opportunity to take a break before continuing.

The end of the experiment was signified by a message with the total number of scored points. Most participants took about an hour to complete the experiment, including the break. The experimenter remained in the room at all times, looking over the participants' shoulder now and then to monitor performance. On request, more information about the study was offered to the participant after completion of the experiment.

3.3 *Pilot study*

After the pilot study with the first 11 participants, it was decided that there should be a break halfway the experiment since participants were complaining about fatigue and their eyes in need of a rest. Because their data would be discarded now, we seized the opportunity to also extend the LD interval in the main experiment from 750ms to 1000ms. This would allow the participants to generate more valid LD trials, since the feedback 'too slow' was shown quite often to the first 11 participants. This was tested later with a t-test, and the first 11 participants significantly missed more LD deadlines than the next 11 participants (out of the 24 in the main experiment) that followed
(t = 2.384, df = 10, p = 0.038).

The desired effect of the break was a stable performance level for the participants during the experiment. Comparing the performance in the blocks before and after the break with a paired T-test revealed no differences in accuracy on the LD task (t = -0.665, df = 23, p = 0.512). Also, the performance in reaction times on the LD task did not differ (t = 1.107, df = 23, p = 0.280), and on the time estimation task as well (t = 0.452, df = 23, p = 0.655).

3.4 *Results*

3.4.1 Outlier definition

We want to examine if there are outliers in our data, so we first look at a plot of the main effects. We plot the accuracy in boxplots (Figure 3.2) averaged over all participants, versus the SOA and the word frequency:

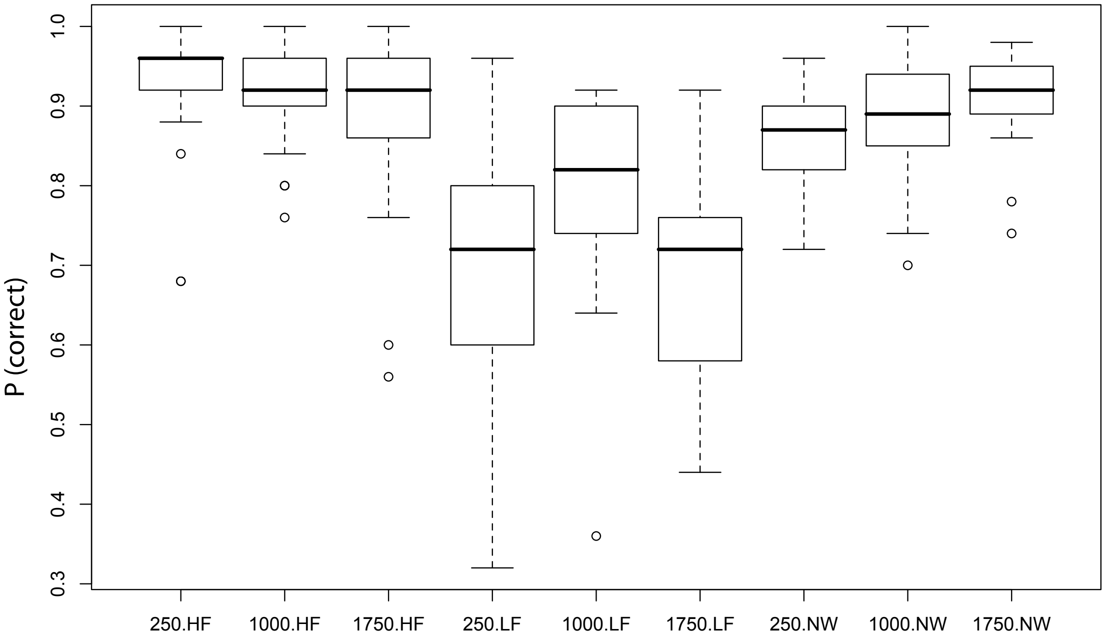

Figure 3.2. Main effects, boxplot for proportion correct scores per SOA per condition.

The middle three boxplots show a peculiar result in the LF condition: accuracy is constant for the 250ms and 1750ms SOA condition, but seems to deviate quite much for the 1000ms condition. One possible cause of this phenomenon is that the words were not randomized for each participant, so that all participants got the same 25 words in the (1000ms, LF) condition. The distributions presented in Figure 3.2 suggest that some words were (too) hard to recognize and decreased performance at 250ms and 1750ms. In theory, the performance on these three SOA's should have a trend in one direction according to our hypothesis, or be similar when there is no effect of SOA on accuracy.

After examining the above phenomenon and in particular the proportion correct scores for each word, we excluded the words for which accuracy scores averaged over participants were 50% or less. The (LF) word with the worst performance ("Dinar") had an accuracy of even 0.167, which means 1 out of 6 trials correct. This exclusion meant removing 17 words in total (5.667% of the LD trials), of which 4 were non-words and 13 were LF words.

Examining the proportion correct scores (Table 3.1), it is very clear that the first three participants (33, 19 and 25) scored much lower on the LD task than the rest of the participants.

We decided to discard data from all participants who scored less than 75% correct on average on the LD task. Therefore, the first three participants were discarded.

Participant	Proportion Correct	Participant	Proportion Correct	Participant	Proportion Correct
33	0.713	22	0.853	18	0.883
19	0.717	13	0.857	28	0.883
25	0.730	34	0.863	31	0.890
14	0.783	27	0.870	15	0.893
26	0.803	35	0.870	24	0.893
32	0.817	21	0.873	12	0.907
17	0.833	23	0.873	30	0.923
16	0.850	20	0.877	29	0.933

Table 3.1. Proportion correct per participant. The first three listed had less than three quarters correct of all the LD answers and can therefore be regarded as outliers.

Next we calculated the standard deviations on the reaction times to see if there were participants which performed much worse than the rest (Table 3.2).

Participant	StDev on RT (TE)	Participant	StDev on RT (TE)	Participant	StDev on RT (TE)
32	259	24	545	20	747
34	269	27	553	14	775
23	312	21	625	18	894
31	315	33	636	29	924
15	368	35	673	17	956
16	471	28	676	26	959
12	512	19	694	25	965
30	538	22	715	13	1140

Table 3.2. Standard deviations on the total reaction time for the TE task.

The standard deviations rise slowly, but the last 6 seem to be much bigger than the rest. We decided that all participants with standard deviations on their RT above 800ms were to be classified as outliers, therefore data from these six participants (18, 29, 17, 26, 25 and 13) was discarded.

3.4.2 Results and discussion for lexical decision

In total we have discarded the data of 8 participants (13, 17, 18, 19, 25, 26, 29 and 33) and the data from 17 words. Discarding data from 1/3 of the participants might seem much but is not uncommon in timing experiments, for example in another timing experiment data of 17 out of 45 participants was discarded (Wearden, 2002).

The results for the proportion correct scores are in Figure 3.3.

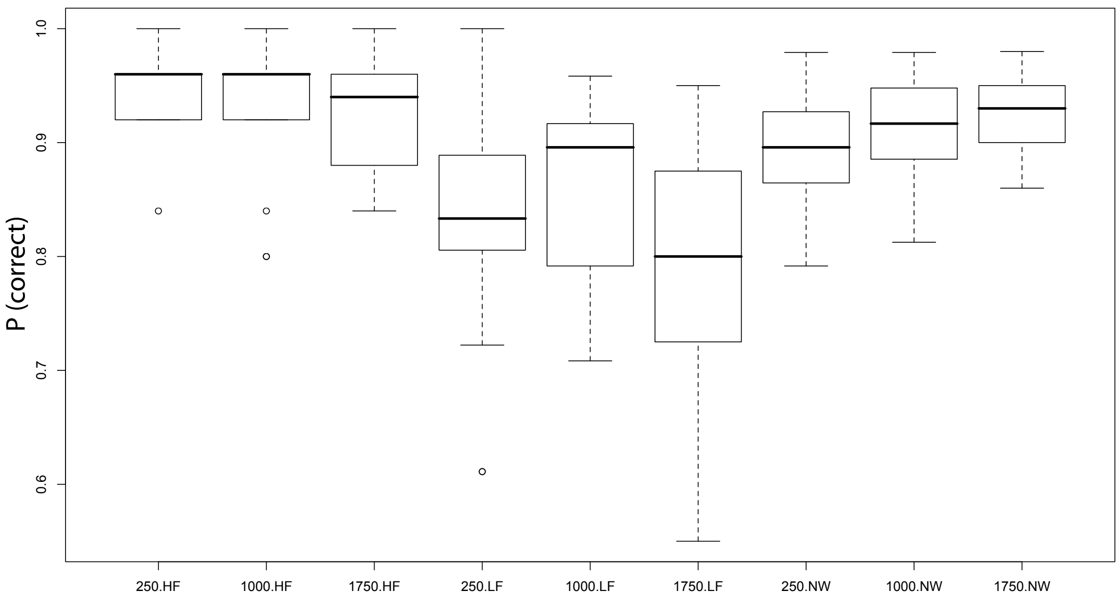

Figure 3.3. Main effects, boxplot for proportion correct scores per SOA per condition.

The HF and NW condition did not change very much comparing to the results from the complete dataset (Figure 3.2), but the strange effect in the LF condition seems less pronounced than before. To test this, we looked at the influence again from SOA on accuracy, per condition. In the overview in Table 3.3 we applied the ANOVA test to all three conditions.

Correlation between	**Significant**	**Measure**
Accuracy and SOA (HF condition) (AOV)	No	F(2,15) = 0.468, p = 0.631
Accuracy and SOA (LF condition) (AOV)	Yes	F(2,15) = 3.862, p = 0.032
Accuracy and SOA (NW condition) (AOV)	No	F(2,15) = 2.798, p = 0.077

Table 3.3. Influence of SOA on accuracy per condition, tested with analysis of variance.

As suspected from the main effects boxplot, in the HF and NW condition there is no relation between SOA and accuracy. In the LF condition however, even after removing 17 words on which performance was bad, there is an effect from SOA on accuracy. Judging from the main effects boxplot, this would mean that accuracy is higher for a SOA of 1000ms than for the other SOA's.

According to our hypothesis, the RT on the LD task should also be dependent on the SOA. Since the threshold builds up faster at low SOA and thus gets larger in the same amount of time, the HF/LF word chunks need more activation to reach the Luce ratio criterion. Therefore, we expect that the RT on the LD task will become larger for low SOA. For the NW condition we would expect the opposite, since this condition profits from a faster rising threshold.

The RT data for the LD task can be seen in Figure 3.4.

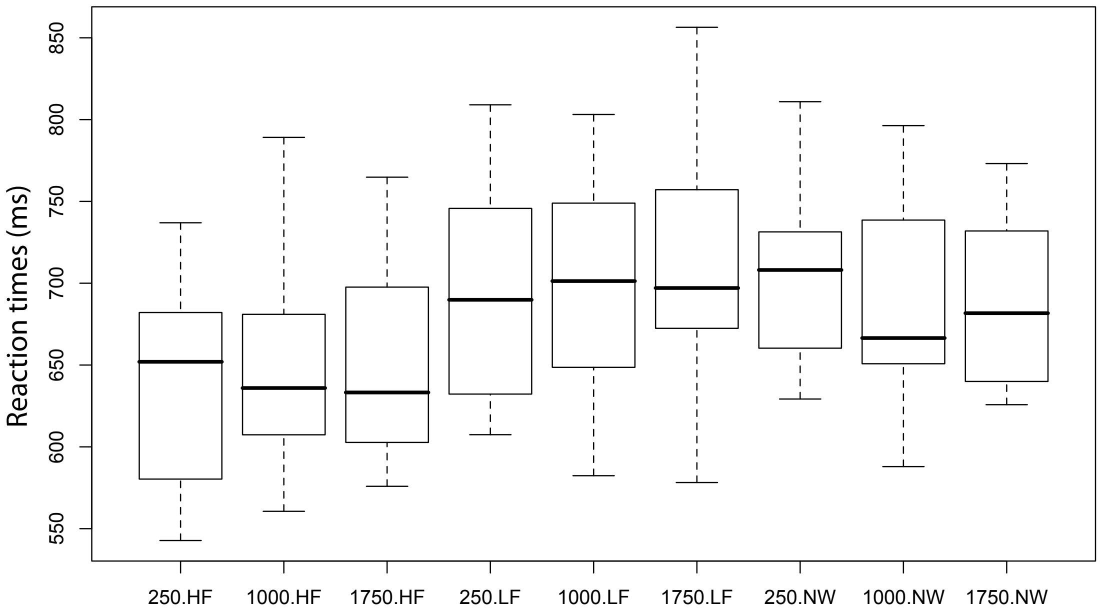

Figure 3.4. Reaction time data on LD task, per SOA per condition.

As mentioned before, reaction time data is ordered HF < LF < NW < VLF, but boundaries between these conditions very strict. Therefore it seems from Figure 3.4 that the LF words might also have been classified as VLF, since at first glance the NW decision seems faster or similar to the LF decision. This is not a problem, on the contrary, because the lower the frequency of the LF words, the slower their activation builds up. This leaves even more room for wrong NW decisions at low SOA, in other words the effect would only be more clear.

We will examine with an ANOVA for each condition (Table 3.4) if our predicted effect from SOA on LD RT is present.

Correlation between	**Significant**	**Measure**
LD RT and SOA (HF condition) (AOV)	No	F(2,15) = 2.242, p = 0.124
LD RT and SOA (LF condition) (AOV)	No	F(2,15) = 2.101, p = 0.140
LD RT and SOA (NW condition) (AOV)	No	F(2,15) = 2.319, p = 0.116

Table 3.4. Influence of SOA on LD reaction time per condition, tested with analysis of variance.

Since there are no significant effects we can conclude that the SOA does not have an influence on the reaction time in LD tasks in our experiment. This result does not support our hypothesis.

3.4.3 Results and discussion for time estimation

When we compare the performance on the time estimation task over all 400 trials, the results can be seen very clearly in Figure 3.5 on the left. The higher the SOA, the longer participants estimate the time interval (F(2,15) = 25.48, p < 0.0001). Even when looking at the first 20 trials instead of the total 400 trials to see if the effect is robust, the effect of the SOA on RT is still present (F(2,15) = 11.85, p = 0.0002).

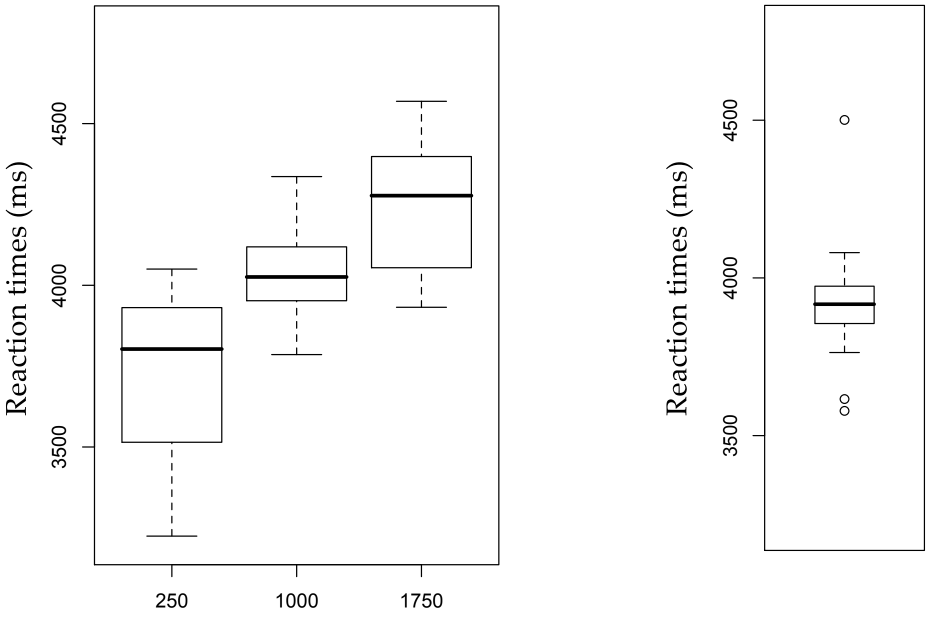

Figure 3.5. Left: Performance on time estimation task per SOA, where a 4 second interval had to be guessed. Right: Performance on the time estimation task without the LD task.

The median in the Figure 3.5 (left) for 1000ms is nearly the defined interval of 4 seconds. We think in this case a learning effect is present, where people tend to have an average number of ticks stored in memory to wait after the LD task, which defines their time interval estimate. We will call this the 'Interval To Wait' (ITW) here, which the participant stores as a number of ticks. Therefore the ITW for the 250ms SOA condition is the largest, and the ITW for the 1750ms SOA condition the smallest.

The reason that this is stored, is that we expect that participants reset their pacemaker (Figure 2.10) after completing the LD task. Since the passed time from the beginning of the interval is unknown at that point, the participant will rely on the ITW for making a time interval estimate.

The ITW that corresponds to the 1000ms condition is most active in the memory because of spreading activation. The ITW's spread activation to their neighbours. The closer the neighbour, which means the less ticks that two ITW's differ, the stronger the spreading activation. The 250ms and 1750ms ITW's both receive spreading activation from the 1000ms ITW and much less from each other, but the 1000ms ITW receives spreading activation from both the 250ms and 1750ms ITW.

Therefore, the 1000ms ITW will be retrieved more of the trials than the other ITW's while all ITW's should be retrieved an equal amount of times.
In the 1000ms SOA condition, this number of ticks is correct. In the 250ms SOA condition, this 1000ms ITW represents a too small interval because it is counted relatively early in the interval. This results in an underestimation of the interval.
In the 1750ms SOA condition, this 1000ms ITW represents a too large time interval and therefore results in an overestimation.

At the right (Figure 3.5) we see the time estimate for the time estimation only task, which is nearly the interval time of 4 seconds. We cannot compare this result with the time estimation of the combined TE and LD trials, since in this case there was no secondary task.

The results on the left (Figure 3.5) correspond to the results from other studies (Van Rijn & Taatgen, 2008), where it is shown that a higher SOA with the same combined task generates a longer estimate of the time interval. Analogous to these results we can explain our own results.
As reasoned above, the participant stores the ITW after the LD task to define the interval, say that the ITW is 14 ticks in the 1000ms condition. Since the 1000ms ITW is most active in memory, this ITW will be retrieved in more cases than it should be. This will happen in a fraction of the 250ms and 1750ms SOA condition trials, depending on the difference in activation between the ITW's. We will use example values here purely to explain our reasoning. The total time interval of 4 seconds takes up 30 ticks, so the length of SOA plus the LD task in the 1000ms SOA condition is 16 ticks.
When SOA is low (250ms), the LD task is already done at 15 ticks and the participant adds 14, counting to 29. This results in a too fast response.
When SOA is at 1000ms, the LD task is done (as predicted) at 16 ticks and the participant counts until 30, adding 14 ticks. In this case, the participant guesses the interval right.
In the case of a high SOA, the LD task and SOA together take up 17 ticks. By adding 14 ticks the participant counts until 31, which results in a too slow response.

The effects as shown above will be stronger in terms of ticks difference between conditions, since SOA's are spaced 750ms apart, but will also be decreased again by the feedback the participant receives on their performance. Also, the ITW's for the 250ms and 1750ms SOA conditions are still stored in memory and although they will be retrieved less of the trials than they should be, they counter the over- and underestimation effect each time they are correctly retrieved. We think a balance is found in between that explains the results found.

We can explain our results both from a RACE perspective by using spreading activation and from a non-linear timing perspective as shown above. Therefore we have good reason to continue to research the influence of timing on decision processes and the non-linear character of time perception.

4 General Discussion

We did not find a significant effect of SOA on accuracy or on lexical decision RT, over all conditions. Therefore our hypothesis concerning time influencing performance on LD tasks cannot be confirmed.
However, lexical decision is a task where a certain amount of uncertainty always remains, because a non-word decision is made with a lack of evidence for the word condition. Therefore, one can never be sure of the NW condition itself.

The absence of the effects of SOA on accuracy could also come from experimental factors. It is possible that by setting a penalty of 25 points on a wrong lexical decision the focus of the participant shifted towards the LD task primarily, even though a correct time estimation still yielded 100 points. The effect of a penalty (negative feedback) might have been more important to a participant than simply missing out on another 100 points (neutral feedback).

All four frequency conditions in our model consist of one chunk in the declarative memory, while in reality the three HF/LF/VLF conditions are categories and contain a lexicon within that category. The NW category logically does not contain a lexicon and is therefore rightly modelled as one chunk. With more chunks within a category, the variance becomes larger. The .9 percentile for the NW condition in our model seems fitted less well than the rest of the data points. However, if the variance in the distribution would increase as just described, the variance in all the word conditions would become larger but the variance in the NW condition would remain the same. Therefore, the model would represent the data even better.
This is in accordance with Zipf's law (Zipf, 1949), which states that the frequency of a word is inversely proportional to its rank in the frequency table. Thus the most frequent word occurs twice as much as the second most frequent word, which in its turn occurs twice as much as the fourth most frequent word, and so on.
This means that occurrences for VLF words have quite a large variance, but that this within-group difference in occurrences is even higher for LF and much higher for HF words. This can also explain very small variance in the distributions of the word conditions in the model.
If we would model the individual words or create a less simplified model than 1 chunk per frequency condition, then it would be useful to compare our LD results with the empirical data mentioned in section 2.4 with respect to the skewness and kurtosis.

In conclusion, we can say that we did not prove our hypothesis that the timing task influences a simultaneously performed lexical decision task. However, in other two-choice tasks with a more binary choice character, this might still be the case. What does become clear is that performing more tasks simultaneously has an effect on timing.

References

Anderson, J. R. (2004). Thoughts on activation and latency.

Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?* New York: Oxford UP.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. L. (2004). An integrated theory of the mind. *Psychological Review, 111*(4), 1036-1060.

Coon, D. (1989). *Introduction to Psychology, Exploration and Application*. St. Paul: West Publishing Company.

Glanzer, M., & Ehrenreich, S. L. (1979). Structure and Search of the Internal Lexicon. *Journal of Verbal Learning and Verbal Behavior, 18*(4), 381-398.

Glaser, W. R., & Dungelhoff, F. J. (1984). The Time Course of Picture Word Interference. *Journal of Experimental Psychology-Human Perception and Performance, 10*(5), 640-654.

Lebiere, C. (2001). *ACT-R 5.0 subsymbolic.* Paper presented at the 2001 ACT-R postgraduate summer school, Berkeley Springs, WV.

Levelt, W. J. M. (1989). *Speaking: From intention to articulation*: Mit Pr.

Link, S. W., & Heath, R. A. (1975). Sequential Theory of Psychological Discrimination. *Psychometrika, 40*(1), 77-105.

Luce, R. D. (1963). Detection and recognition. In R. D. Luce, R. R. Bush & E. Galanter (Eds.), *Handbook of Mathematical Psychology* (I ed., pp. 103-189). New York: Wiley.

Matell, M. S., & Meck, W. H. (2000). Neuropsychological mechanisms of interval timing behavior. *Bioessays, 22*(1), 94-103.

Mccormack, P. D., & Wright, N. M. (1964). Positive Skew Observed in Reaction-Time Distributions. *Canadian Journal of Psychology, 18*(1), 43-&.

Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance .1. Basic mechanisms. *Psychological Review, 104*(1), 3-65.

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard UP.

Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review, 85*(2), 59-108.

Schneider, W., Eschman, A., & Zuccolotto, A. (2002). *E-Prime: User's Guide*: Psychology Software Inc.

Sun, R. (2006). The CLARION cognitive architecture: Extending cognitive modeling to social simulation. In *Cognition and multi-agent interaction*. New York, NY: Cambridge UP.

Taatgen, N. A., van Rijn, H., & Anderson, J. (2007). An integrated theory of prospective time interval estimation: The role of cognition, attention, and learning. *Psychological Review, 114*(3), 577-598.

Usher, M., & McClelland, J. L. (2001). The time course of perceptual choice: the leaky, competing accumulator model. *Psychological Review, 108*(3), 550-592.

Van Maanen, L., & Van Rijn, H. (2007). An Accumulator Model of Semantic Interference. *Cognitive Systems Research, 8*(3), 174-181.

Van Maanen, L., & Van Rijn, H. (2008). *The picture-word interference effect is a stroop effect after all.* Paper presented at the 30th Annual Meeting of the Cognitive Science Society, Washington DC.

Van Maanen, L., Van Rijn, H., & Taatgen, N. A. (subm.). Accumulators in context: An integrated theory of context effects on memory retrieval.

Van Rijn, H., & Anderson, J. R. (2003). *Modeling Lexical Decision as Ordinary Retrieval*. Paper presented at the Proceedings of the Fifth International Conference on Cognitive Modeling.

Van Rijn, H., & Taatgen, N. A. (2008). Timing of multiple overlapping intervals: How many clocks do we have? *Acta Psychologica, 129*(3), 365-375.

Vickers, D., & Lee, M. D. (1998). Dynamic models of simple judgments: I. Properties of a self-regulating accumulator module. *Nonlinear Dynamics, Psychology, and Life Sciences, 2*(3), 169-194.

Wagenmakers, E. J., Ratcliff, R., Gomez, P., & McKoon, G. (2008). A diffusion model account of criterion shifts in the lexical decision task. *Journal of Memory and Language, 58*(1), 140-159.

Wagenmakers, E. J., Steyvers, M., Raaijmakers, J. G. W., Shiffrin, R. M., van Rijn, H., & Zeelenberg, R. (2004). A model for evidence accumulation in the lexical decision task. *Cognitive Psychology, 48*(3), 332-367.

Wearden, J. H. (2002). Traveling in time: A time-left analogue for humans. *Journal of Experimental Psychology-Animal Behavior Processes, 28*(2), 200-208.

Zeelenberg, R., Wagenmakers, E. J., & Shiffrin, R. M. (2004). Nonword repetition priming in lexical decision reverses as a function of study task and speed stress. *Journal of Experimental Psychology-Learning Memory and Cognition, 30*(1), 270-277.

Zipf, G. (1949). *Human Behavior and the Principle of Least Effort*. Cambridge, MA: Addison-Wesley.