Modeling Lexical Decision as Ordinary Retrieval

Hedderik van Rijn (hedderik@cmu.edu)
John R. Anderson (ja@cmu.edu)
Department of Psychology
Carnegie Mellon University

Abstract

We present an ACT-R model of signal-to-respond lexical decision. This model is based on a mental lexicon constructed from a complete list of English four-letter words with associated word frequencies. A number of (signal-to-respond) lexical decision phenomena are explained by the model, without having to rely on mechanisms specific to lexical decision. Besides explaining these phenomena, the model also illustrates the necessity of a competitive latency mechanism. According to this mechanism, the time necessary for retrieving an element from declarative memory is a function of the activation of all other elements in declarative memory. We argue that the data presented here can only be explained using a competitive latency mechanism.

General Introduction

A task often used to study the human mental lexicon is lexical decision. In a lexical decision task, participants are presented strings of letters that have to be categorized as word (e.g., "tree") or nonword (e.g., "arda"). Experiments involving lexical decision are generally aimed at shedding light on the issues related to the storage and retrieval of words during normal linguistic processing. However, most accounts of human lexical decision performance involve task-specific mechanisms that are irrelevant to normal linguistic processing. The model presented in this paper proposes an explanation for lexical decision that solely relies on mechanisms involved in everyday memory processing.

This paper focuses on the empirical data reported by Wagenmakers et al. (2001, Exp. 2), parts of which were earlier published in Steyvers, Wagenmakers, Shiffrin, Zeelenberg, and Raaijmakers (2001). The work presented by Wagenmakers et al. focuses on three phenomena: (1) The *word frequency effect*. Words that occur more often are recognized faster and more accurately than words that occur less often. (2) The *repetition priming effect*. If a word has been presented recently, second presentations are associated with faster correct classifications. However, nonwords are often reported to have a decreased accuracy on second presentation. (3) The *nonword lexicality effect*. Nonwords that are more like legal words (e.g., by following the orthographic rules of a language more closely) take longer to be classified than nonwords that deviate more substantially from legal words.

Participants in classical respond-when-ready (or "speeded") paradigms of lexical decision are typically instructed to respond "as fast and accurately as possible". Besides introducing the classical speed-accuracy trade-off, the respond-when-ready paradigm also limits the information available for inspection to one data point after processing has been finished. As the dynamics of processing might be different between conditions that eventually result in similar data, it can be difficult to distinguish between different accounts of behavior. In a signal-to-respond task setting (Antos, 1979; Hintzman & Curran, 1997), participants are required to respond at an (audible) signal. By distributing a number of signals over a typical lexical decision reaction time, intermediate accuracy information becomes available that reflects the time course of processing.

In their study, Wagenmakers et al. presented participants[1] four types of stimuli of either four or five letters in a signal-to-respond lexical decision task: 168 high frequent (HF) words, 168 low frequent (LF) words, 168 pronounceable nonwords (NW1) that had been constructed by replacing one letter from a word, and 168 pronounceable nonwords (NW2) that differed at least two letters from a word. Because of modelling constraints, we will focus in this paper on the four-letter words, leaving 48 words per condition. The time (lag) between presentation of the stimulus and the respond signal was 75, 200, 250, 300, 350, or 1000 milliseconds. Each of the stimuli was presented twice, and after each presentation, accuracy and timing feedback was provided to the participant.

The four-letter word results for both the initial presentation (open markers) and repeated presentation (closed markers) are presented in Figure 1 as a function of lag. The y-axis represents the proportion of word answers to the stimuli presented for the various conditions. As this axis represents the answer for the correct response for the word conditions but represents the incorrect response for the nonwords, this figure can be thought of as a hit/false-alarm plot. The data points at the right side of this figure, at a lag of 1000 milliseconds, resemble data commonly found in respond-when-ready lexical decision tasks: high

[1] Note that a disadvantage of the signal to respond method is its difficulty from the participants' perspective. It is not uncommon to have to remove the data of 25% of all participants because of poor response timing, excess errors, etc.

accuracy for HF and NW2 conditions, lower accuracy for LF and NW1 conditions

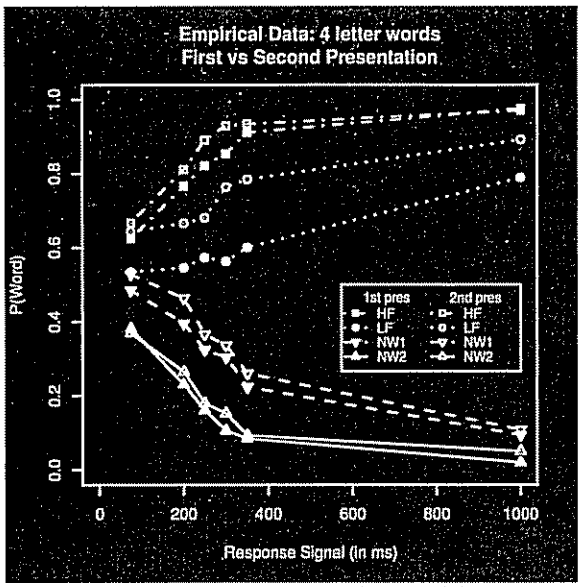

Figure 1 Empirical observed hit and false-alarm rates for the subset of four letter words in the signal-to-respond experiment as reported in Wagenmakers et al (submitted, Exp 2) Note that the data is plotted based on the signal time, not on the actual reponse times See text for the explanation of the abbrevations

The most striking effect shown in this figure is the effect of processing time on the P(W) for both word and nonword conditions: the more time is given for a response, the higher the accuracy (i e , the higher the chance of a word-answer for word stimuli and the lower the chance of a word-answer for nonwords) However, whereas both the HF and NW2 condition show a fast increase in accuracy, performance for the LF condition is roughly at chance for the first five lags, only to improve for the longest lag The NW1 condition takes an intermediate position, dropping not as fast as the NW2 condition, but also not remaining as long at chance level as the LF condition

The effect of repetition priming is straightforward for both word conditions: the second presentation shows improved accuracy Although this effect is relatively small for HF words - probably due to ceiling effects - LF words show a large effect of repetition priming, almost resembling the accuracy for HF words at first presentation With respect to the nonword conditions, the accuracy decreases over time for both NW1 and NW2 stimuli, but the decrease of the NW2 condition is relatively small compared to the decrease of the NW1 condition

The increase in accuracy for the word conditions might be explained by assuming instance-based memories, linking the stimuli to the feedback given at the end of each trial However, the significant decrease in performance for the NW1 condition refutes this explanation That is, if instances formed at the first presentation would be used to answer the second presentation; the NW1 and NW2 conditions should also have shown an increased accuracy instead of the observed decrease (Steyvers et al , 2001)

The ACT-R Model of Lexical Decision

ACT-R

ACT-R (Anderson & Lebiere, 1998) is a hybrid architecture in which production rules control behavior This behavior is often mediated by the availability of declarative knowledge The current discussion is mainly focused on declarative memory, as this part of the architecture underlies the model's explanations of behavior Declarative memory in ACT-R consists of typed memory structures, referred to as chunks Each type of chunk has a predetermined number of slots, representing, for example, the individual letters of a word Furthermore, each chunk has an associated activation (B_i), calculated as:

$$B_i = \ln\left(\sum t_j^{-d}\right)$$

This function determines the relative contribution of a previous encounter (j) to the base-level activation of chunk i as a power function of the time (t_j^{-d}) that has passed since that encounter The decay parameter (d) is traditionally set at .5

When a production rule requests the retrieval of a chunk, the activations of all chunks are updated to reflect the context of the current retrieval request As B_i reflects the activation of a chunk when the context fully matches, this updating involves accounting for mismatches between context and chunk slots The activation of a chunk can therefore be expressed as:

$$A_i = B_i - \sum D_j + \sigma$$

where D is a fixed penalty if the letter in slot j mismatches The last term in this formula, σ, reflects the moment-to-moment normal distribution based noise added to the activations (mean of 0, SD of 0.5) After this updating, the most active chunk is selected for further retrieval One of the competitors for retrieval is the retrieval threshold (τ) That is, if none of the chunks has an activation above τ, the threshold is "retrieved", signaling a failed retrieval

In contrast to the default ACT-R chunk retrieval latency mechanism, the current model uses a competitive latency mechanism (Lebiere, 2001) This change from a local latency mechanism to a global, competitive latency mechanism will be extensively discussed in the Discussion According to this competitive latency mechanism, the time necessary to retrieve the most active chunk (i) depends both on its own activation (A_i), the activation of all other chunks (A_j), the retrieval threshold (τ), and two scaling parameters (F and f, of which F was fixed at 1):

$$\text{Time}_i = F\left(\frac{\sum e^{A_j} + e^{\tau}}{e^{A_i}}\right)^{\gamma}$$

The Model

The model hypothesizes that the behavior of a participant in the lexical decision task consists of the following steps:

1. The presented letter string is encoded in a production rule that initiates the retrieval request
2. The procedural part of the model waits for an event to occur, being either a signal-to-respond or the finished declarative retrieval.
3. If a word is retrieved before the signal sounds, note that the answer is "word" and wait for the signal before pressing the word-button
4. If the declarative system reports that all words are less active than the retrieval threshold, note that the answer is "nonword" and wait for the signal before pressing the nonword-button.
5. If the signal-to-respond is perceived after a word/nonword decision has been made based on information from the declarative memory, report that decision.
6. If the signal-to-respond is perceived before the retrieval is finished, guess the answer by choosing between word and nonword with a pre-specified ratio.

Assuming that the encoding of the stimulus and the response processes take a fixed amount of time, the main determinant of behavior is the retrieval process, initiated at Step 1 and finalized at either Step 3 or 4. The data shown in Figure 1 plots the P(W) against the signal times, not against the (average) reaction times. The average difference between the actual response and the signal is 216 ms, which is roughly similar what would have been predicted by the perceptual/motor interface of ACT-R for the encoding of a string and the pre-programmed pressing of a button.

Traditionally, ACT-R models rely on a relatively small number of declarative facts that are learned during the course of a simulated experimental session. However, a model of lexical decision necessarily requires the incorporation of a mental lexicon that represents the aggregation of long-term experience with word recognition. Although literature reports several successful models with mental lexicons of limited size and artificial lexicons, the model discussed here is based on the complete CELEX four-letter word lexicon of English (Baayen, Piepenbrock, & Van Rijn, 1993). That is, each word in the original database is represented in declarative memory as a memory structure with four slots, each of these slots representing a position-encoded letter. The total number of four-letter words in the CELEX database is slightly less than 7000. However, for simplicity we assume that homographs share their orthographic representations, bringing the number of chunks in the model down to 2479. Each of these words has an associated base-level activation, determined on the basis of its CELEX wordform frequency (in the case of the homographs, based on their summed frequency) which is added to a base frequency of 50. As no information is available about the distribution of word encounters, we assume an evenly spaced distribution of encounters. Given this assumption, the base-level activation is determined by the optimized version of the base-level activation equation:

$$B_i = \ln\left(\frac{n}{1-d}\right) - d\ln(T)$$

In this equation, n equals the word's frequency and T represents the total age of the chunk. As we have no information of the age of individual chunks, this value is approximated for all chunks by the sum of all word frequencies (i.e., $T=\sum(n_i)$).

If a word was retrieved during the first presentation, its baselevel activation is increased by the ACT-R learning mechanism. Based on the details of the task of the experiment we assumed 48 seconds between first and second presentations[2]. The optimized activation formula used for the frequency based activation and the repetition priming activation component are combined by: $B_{2nd}=\ln(e^{B_{1st}}+B_i)$, in which B_{2nd} is the activation at the start of the second presentation, B_{1st} is the activation based on the CELEX frequency, and B_i is the activation related to the recent encounters (see Petrov & Anderson, 2000, for a similar approach).

This process is straightforward to apply to the items in the word conditions. If a word is retrieved, its number of references and therefore its activation is updated. With respect to the nonword conditions, we assume that if a "word" answer is given, the nonword might have been incorporated in the declarative memory in a representation similar to the "normal words". These new representations start out with a frequency equal to the frequency of the lowest frequent word.

Taken together, the assumptions underlying the repetition priming in the current model is that when a word answer is given, a word representation is primed, either by increasing the activation of an existing word, or by the construction of a new chunk representing the presented stimulus.

Simulation Results

Figure 2 depicts the simulation results of the above-discussed model[3]. To average out the effects of noise in activation, this figure shows the averaged results of 10 model runs. The lines with the filled markers represent

[2] As a retrieval often involves the subsequent representation of that chunk in a production rule, a single retrieval is typically represented as two closely spaced references. In the model presented in this paper, both t values are set at 48 seconds.

[3] An R simulation (Ihaka & Gentleman, 1996) of this model is available at http://www.van-rijn.org/hedderik/storld

the first presentation data; the lines with open markers represent the second presentation data.

If the model encountered the signal-to-respond before having any other information available, it guesses an answer. The proportion of word answers when guessing was set at .44, being the mean P(W) of the LF words and the two nonword conditions at the first lag. As the HF word condition already seems to have an effect on P(W) at the shortest lag, this condition was not included in setting the guessing parameter.

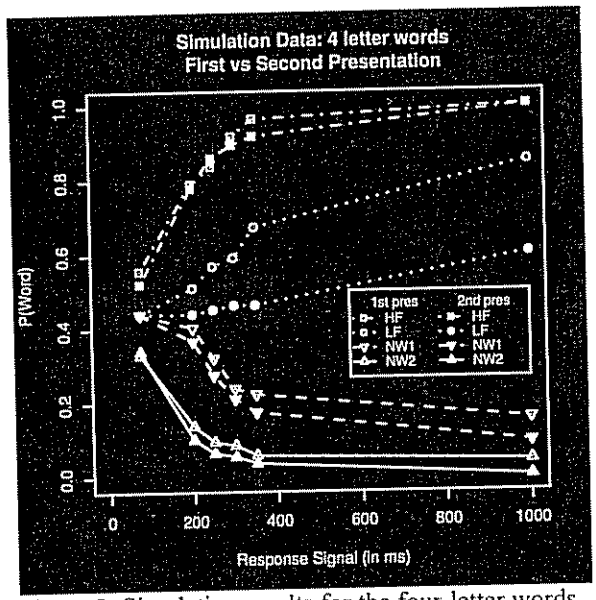

Figure 2. Simulation results for the four-letter words.

For the data of the first presentation, the *f* parameter was set at 0.5 to constrain the effects of the large number of chunks on the competitive latency. The mismatch penalty was estimated at $D = 6$ and the retrieval threshold at $\tau = -3$. Based on these estimations, the model's fit of the first presentation data is $R^2 = .97$. For the second presentation's data, no parameters have been changed nor are new parameters introduced. As discussed earlier, references to words in the initial presentation increase the words activation by a value directly derived from the experimental setup. The model's fit of the second presentation is $R^2 = .94$, resulting in an overall R^2 of .95.

The answers given by the model can be either guess based (P(W)=.44), based on the retrieval threshold (P(W)=0), or based on the retrieval of a correct or an incorrect word (P(W)=1). The breakdown of the behavior of the model for the first presentations in terms of these underlying mechanisms is shown in Figure 3. This figure shows the underlying dynamics of the simulation depicted in Figure 2. At short lags, specifically for the NW1 and LF words conditions, the answer mechanisms have not had the chance to influence the answer decision, yielding the majority of answering being based on guessing, which results in a P(W) of about .44. At longer lags, the declarative memory system has enough time to return information, either being a retrieved word or representing a retrieval failure. Together, the word retrievals and the retrieval threshold determine the accuracy.

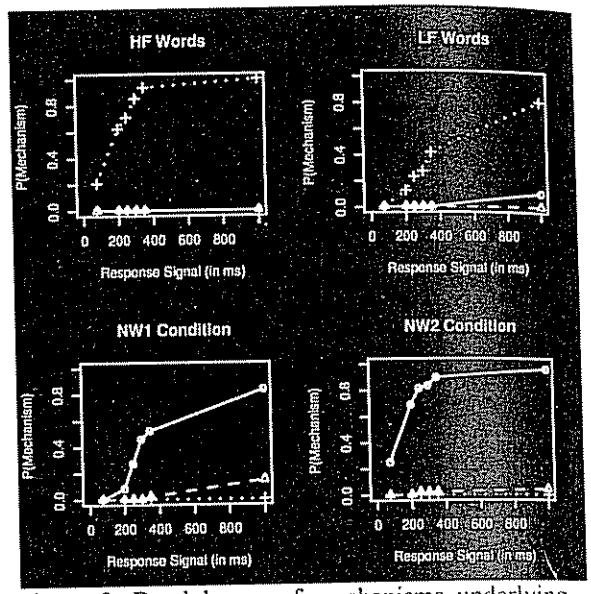

Figure 3. Breakdowns of mechanisms underlying behavior for the first presentation simulations. Solid lines marked with circles represent the proportion of answers based on the retrieval threshold, dashed lines marked with triangles represent retrieval of another word than the target word, and dotted lines with plusses represent the retrieval of the target word. Not shown are the guesses whose proportion is 1 minus the observed proportions.

Figure 3 illustrates that most of the answers at long lags are based on the (correct) retrieval of the target word, or on the inability to retrieve a word. Only in the NW1 condition, the model retrieves an incorrect word in a number of cases. As the model does not check the retrieved chunk against the perceived input, it simply answers word based on this retrieval. The figure also shows that for each condition one mechanism dominates the others; the time course of this mechanism is the main predictor of the behavior of the model. For both the HF words and the NW2 conditions, the relative contribution of the involved mechanisms quickly rises to almost 100% usage with very little guessing. For both the LF words and the NW1 conditions, the performance rises more gradually. As both the retrieval of a word and the retrieval threshold mechanisms take longer, behavior is close to chance level for a longer period. This is caused by the combination of two processes. As the activation of the LF words is relatively low, their latencies are relative long. When it takes longer to retrieve something from memory and therefore to base an answer on retrieval, the guess mechanism determines behavior for a longer period.

The other process is related to the competitive latency mechanism. As the ratio between the targeted chunk (or the retrieval threshold) and the other chunks is small, it will take longer to retrieve that information, yielding a similar effect as described above. For the LF words, this ratio is small because of their relative low frequencies. For the NW1 condition, the latency for the retrieval threshold is increased because of the higher number of similar word competitors ("neighbors").

With respect to the repetition priming for the word conditions, because of the new additional references to the word chunks, their P(W) increases. However, as ACT-R predicts a greater impact of additional references on the activation of low activated words (i.e., low frequency words) than on the activation of high activated words, the model shows an increased effect for the LF words compared to the HF words. The ceiling effect discussed in the context of Figure 1 is therefore not purely based on a ceiling in the P(W) statistic, but would also exist based on the leveling properties of the ACT-R activation and retrieval equations as an additional reference would not add much activation in absolute terms to an already highly active chunk. The repetition priming of the nonword conditions is explained by the assumption that the "word" answers to nonwords can lead to representing that nonword in memory. If the model retrieves the representation of that nonword at the second retrieval, the model answers "word". Therefore, P(W) will also increase for the nonword conditions. As the number of word responses during the first presentation is lower for the NW2 condition than for the NW1 condition, the increase in P(W) will be smaller for NW2 than for NW1.

Discussion

In the introduction, three phenomena were identified that are often associated with lexical decision. The model presented in this paper can explain all three phenomena relying solely in terms of mechanisms that would play a role in normal word recognition or memory.

The explanation for the *word frequency effect* is most straightforward. The observed empirical frequencies of all words are used to determine their activations. As the activation of a chunk is the main determinant of the time necessary to retrieve that chunk, the word frequency effect is easily explained. However, the model does show that (1) empirically gathered frequencies are good inputs for determining the activation of chunks representing words; and (2) the model shows correct behavior for both frequency conditions, even when a complete, corpus-based lexicon is included, and does not have the scaling-up problems as often associated with computational models (although representational issues remain, as the current model only incorporates four-letter words). With respect to the first issue, the model predicts both the effects for the HF and LF words using the raw frequencies (with the additional base frequency). No further transitions were necessary, and because the activation is directly based on the frequency, no parameters have to be used to define the relation between the HF and LF words (c.f., Wagenmakers et al., 2001). An additional advantage of using a complete corpus as mental lexicon is that it provides a direct way to assess the behavior of (individual) words. This provides an opportunity to eventually compare empirical data and the behavior of the model at a finer grain size. The second issue also illustrates the robustness of the ACT-R implementation of this model. By including a complete lexicon, the model avoids issues such as scaling-up problems or brittleness. Furthermore, this approach also allows easy conversion to other corpora. The model can be equipped with a different lexicon without concerns about necessary extensive retraining, as long as a sensible estimate is known for each word's frequency.

The *repetition priming effect* is explained by increasing the number of references (i.e., the frequency) to a chunk. Again, the basic ACT-R mechanisms suggested a straightforward implementation of this phenomenon that proved to be highly successful in that the model's repetition priming predictions are essentially a zero-parameter fit. Given the nature of the activation formulae, the relative increase in activation for a HF chunk is smaller than the relative increase in activation for a LF chunk. Because of this negative correlation between the amount of additional activation from priming and the before-priming activation of the primed chunk, ACT-R does inherently predict the larger priming effects for LF words.

Without the competitive latency equation, the nonword lexicality effect could probably not have been explained within the ACT-R framework. Given the importance of this change, it will be discussed in more detail below.

Competitive Latency and Nonword Lexicality Effects

When using the default ACT-R latency equation, the retrieval latency of a chunk is a direct function of the activation of that chunk (A_i) scaled by two parameters (F and f, of which the latter is often fixed at 1):

$$\text{Time}_i = Fe^{-fA_i}$$

The time to register a retrieval failure is determined by the retrieval threshold (τ), replacing the activation (A_i) with the threshold's value. According to this formulation, the time to register a retrieval failure is solely based on the activation of the threshold itself. As the threshold is obviously set to a fixed value for all levels of an experiment, no difference in behavior can be observed for conditions in which behavior is mainly determined by the retrieval threshold. Therefore, observed differences in the nonword conditions as shown in Figure 1 cannot be explained using the default ACT-R latency equation to determine the latency of a retrieval failure. A solution that seems to be straightforward is to assume that because of the relative

higher word resemblance of NW1 stimuli than of NW2 stimuli, the model retrieves more often words similar to the NW1 stimuli This way, the P(W) would increase if one assumes a word retrieval results in a "word" response. However, as the difference in P(W) for the NW1 and NW2 condition starts at very short lags, the retrievals of similar words have to be about as fast as in the HF word condition And although the median of the frequencies of the words most similar NW1 nonwords is higher than that of the LF words (585 vs 69), it is not close to the median frequency of the HF words (1323) Even if this difference would cause P(W) to be increased for the short lags, it would also assume that the relative difference between the P(W) ratios for NW1 and NW2 would increase over lags That is, the longer the lag, the higher the chance that a similar word has been retrieved As Figure 1 shows, the difference between the P(W) ratios for NW1 and NW2 does not become bigger (In fact, the difference shows an inverted U-shape, being small at the short and long lags, and bigger at the intermediate lags) Therefore, it seems that there is no straightforward way to explain the results for the NW1 and NW2 conditions using the default latency equation and the assumption that the retrieval of a word always results in a "word" reponse. If one would assume that the retrieval of an incorrect word yields a "nonword" respone, an explanation using the default latency equation becomes even harder. As the NW1 stimuli share more features (i.e., letters) with existing words, these similar words will be retrieved more frequently for NW1 stimuli than for NW2 stimuli. As the retrieval of an incorrect word yields a decrease in P(W), this would predict P(W) for NW1 to be lower than for NW2

Using the competitive latency equation combined with the when a word is retrieved, give "word" as response notion provides an intuitive solution to this problem As has been described earlier, this mechanism states that the retrieval latency of a chunk (or the retrieval threshold) is a function of the activation of all other chunks in declarative memory Because NW1 stimuli are more like "normal" words, these stimuli share more features with words in the mental lexicon than NW2 stimuli. Therefore, the total mismatch penalty administered is smaller in the case of an NW1 stimulus trial than in a NW2 stimulus trial. Because of the higher overall activation (caused by less decrease), the numerator of the competitive latency equation is bigger, resulting in slower retrievals. As the retrieval threshold can be thought of as a chunk that competes for retrieval, its latency will also be increased. Therefore, the onset of responses based on the retrieval threshold will be delayed, causing the model to guess its answer for a longer period. As guessing results in higher P(W) values, the NW1 condition will show higher false-alarm rates.

Acknowledgments

The research reported in this paper was supported by ONR Grant N00014-96-1-0491.

References

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Antos, S. J. (1979). Processing facilitation in a lexical decision task. *Journal of Experimental Psychology: Human Perception and Performance,* 5(3), 527-545.

Baayen, H., Piepenbrock, R., & Van Rijn, H. (1993). *The CELEX lexical database (CD-ROM)*. Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA.

Hintzman, D. L., & Curran, T. (1997). Comparing retrieval dynamics in recognition memory and lexical decision. *Journal of Experimental Psychology: General,* 126(3), 228-247.

Ihaka, R., & Gentleman, R. (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics,* 5(3), 299-314.

Lebiere, C. (2001). ACT-R 5.0 subsymbolic computations. In *Proceedings of the 2001 ACT-R post-graduate summer school*. Berkeley Springs, W. Va. (Available at http://act-r.psy.cmu.edu/workshops/Workshop-2001/schedule)

Petrov, A., & Anderson, J. R. (2000). Fitting the ANCHOR model to individual data: A case study in Bayesian methodology. In *Proceedings of the twenty-second annual conference of the cognitive science society* (pp. 369-374).

Steyvers, M., Wagenmakers, E.-J. M., Shiffrin, R. M., Zeelenberg, R., & Raaijmakers, J. G. W. (2001). A Bayesan model for the time course of lexical processing. In *Proceedings of the fourth international conference of computational modeling*.

Wagenmakers, E.-J. M., Steyvers, M., Raaijmakers, J. G. W., Shiffrin, R. M., Van Rijn, H., & Zeelenberg, R. (2001). *A Baysian model for lexical decision*. Manuscript submitted for publication.