Building an ACT-R reader for eye-tracking corpus data

Jakub Dotlačil (jdotlacil@gmail.com)
Institute for Logic, Language and Computation, University of Amsterdam
Amsterdam, the Netherlands

Abstract

ACT-R has been successfully used in psycholinguistics to model processing data of individual experiments. In this paper, I show how it could be scaled up to model a much larger set of data, eye-tracking corpus data. It is shown that the resulting model has a good fit to the data for the considered (low-level) processes. The paper also argues that free parameters of ACT-R could and should be estimated using the well-established methods in other fields, rather than by manually searching through parameter space. The latter option is simply impossible to use once we hit the amount of data considered here. The latter option also makes it hard, if not impossible, to compare parameters across different (ACT-R) models since manual search is subjective and usually not well documented in research papers.

Keywords: parsing; eye tracking; modeling eye tracking; ACT-R; modeling eye-tracking corpus data; Bayesian inference of ACT-R parameters

Introduction

ACT-R (Adaptive Control of Thought–Rational) (see (Anderson, Bothell, & Byrne, 2004) for an introduction) is a cognitive architecture that has been successfully applied to various language processing phenomena, for example, syntactic parsing, memory retrieval of arguments and quantifiers, syntactic priming or the reanalysis of syntactic structures (Lewis & Vasishth, 2005; Vasishth, Bruüssow, Lewis, & Drenhaus, 2008; Reitter, Keller, & Moore, 2011, a.o.). The successes of ACT-R in modeling natural language strongly suggests that the cognitive architecture can be insightful for linguistics, alongside many other domains of inquiry (cf. (Anderson, 2007)).

Previous applications of ACT-R focused on modeling of (some) results of carefully chosen experiments. This leaves open the question as to how ACT-R fares once we move beyond such a domain. If ACT-R is to be useful for language modeling it should be shown that it can scale up, that is, it can fare well when modeling a large amount of processing data (cf. (Taatgen & Anderson, 2002) for such a large scale ACT-R application in a different psycholinguistic domain). Furthermore, it is important to see how it fares when modeling data that are naturally occurring, not carefully composed by experimentalists to target one phenomenon. Second, previous models were hand-crafted to match analyzed phenomena. This can be seen in two ways: (i) grammar rules are not created automatically, rather, they are manually written, (one exception here being (Reitter et al., 2011)) (ii) parameters used in the sub-symbolic part of ACT-R are plugged in by modelers.

In this paper, I will focus on the second issue: the manual search of parameters. The problem with that is that it makes model fitting subjective. As a consequence, it is very hard if not impossible to compare various models. For example, (Vasishth et al., 2008) differ from (Lewis & Vasishth, 2005) in the values they assume for the latency factor (0.46 vs. 0.14). The model in (Reitter et al., 2011) differs from both papers in its assumption about the value of the maximum associative strength (50.0 in the latter vs 1.5 in the former papers). It is not clear whether these differences are meaningful or accidental. We do not know how good the model fit would be if the values of these parameters were matched. We also do not know what values were considered before settling on these. Finally, we also do not know whether other parameters were searched before these were modified. All these concerns make it hard, if not impossible, to consider model comparisons. Maybe even more importantly, selecting the values of parameters by hand is almost impossible once we scale up and model more data, especially if we want to fit more than one parameter.

In this paper, I take first steps to address the worries discussed above. Further improvements should follow in the future. First, I consider the application of an ACT-R parsing model to eye-tracking *corpus* data (the GECO corpus, (Cop, Dirix, Drieghe, & Duyck, 2016)). Second, I show how the model can be fitted using Markov-Chain Monte Carlo (MCMC) methods, rather than a manual selection of parameters. Importantly, using MCMC methods makes it easy to compare the parameters of the current model to other models. As an example, I make one such comparison, which will reveal a match between some (but not other) parameters, potentially opening a window into more detailed research into the role of ACT-R free parameters across models.

Modeled data

The paper presents a model of (a subset of) reading measures of the Ghent Eye-Tracking Corpus, GECO (Cop et al., 2016). The corpus consists of eye movement measures collected during reading of the book *The Mysterious Affair at Styles* by Agatha Christie. The data were collected from 14 English monolingual readers and 19 Dutch-English bilingual readers. For the current purposes, we are not interested in the effect of bilingualism and thus, only monolingual data will be studied.

A desirable feature of the GECO is that the whole corpus is freely downloadable and its text is in the public domain. Furthermore, the fact that readers read an entire book, rather than the collection of random articles/sentences might potentially be useful in the future if we want to model long-lasting effects (e.g., discourse structures). However, this will not be attempted here. For the details of the corpus and its comparison to other eye-tracking corpora, see (Cop et al., 2016).

I am afraid I showed my surprise
rather plainly.

Figure 1: Example of a parsed sentence.

$$\text{AFRAID}\begin{bmatrix}\text{ISA:} & \text{word}\\ \text{FORM:} & \text{afraid}\\ \text{CATEGORY:} & \text{adjective}\end{bmatrix}$$

Figure 2: Example of the chunk AFRAID.

Basic ACT-R reader

The reader considered in this paper is very basic. It serves as the starting point and it can be further expanded.

The reader will visually encode and retrieve words from English sentences of the GECO corpus, an example of which is in Fig. 1 (the figure encodes the original line breaks).

The reader starts at the first word of the sentence. It stores the word in its visual buffer and retrieves information about the word from its mental lexicon. Once retrieved, the reader shifts its focus to the next word of the sentence, repeating the process. When getting to the end of the line (the word *surprise*), the reader shifts its visual focus to the beginning of the next line and proceed in reading. After the last word of the sentence, the first word of the next sentence will be parsed.

Obviously, the reader in its current form is primitive. It models only visual processes present in reading and processes tied to lexical retrieval. This limitation is intentional. It is important to show that even such primitive models are tangible and useful in modeling eye-tracking corpus data. Once the model is in place, we can move to more complex cases.

Details of the model

Symbolic part As is well-known, ACT-R subsumes two types of knowledge: declarative knowledge and procedural knowledge (cf. (Newell, 1990) on the difference). While the declarative knowledge represents our knowledge of facts, procedural knowledge is knowledge that we display in our behavior (cf. (Newell, 1973)). Following all previous works on ACT-R processing I will assume that lexical information is part of our declarative knowledge. In contrast to that, reading itself is part of our procedural knowledge. The reading consists of finding a word, retrieving the information about the word from the declarative memory and moving one's attention from word to word (in the left-to-right, top-to-bottom fashion).

The declarative knowledge is instantiated in chunks. The procedural knowledge is instantiated in production rules (productions for short).

The chunks storing lexical knowledge can be kept simple, given the basic aims of the presented ACT-R reader: they only store the information about the form and its category, see Fig. 2.

The procedural knowledge consists only of a handful of rules, shown in Fig. 3 to Fig. 6.

```
=g >
state                start
=visual_location >
?visual >
state                free
buffer               empty
==>
=g >
state                retrieve
+visual >
cmd                  move_attention
screen_pos           =visual_location
```

Figure 3: Rule ATTEND WORD.

```
=g >
state          retrieve
=visual >
value          =val
?retrieval >
state          free
==>
=g >
state          shift
word           =val
−visual >
+retrieval >
form           =val
```

Figure 4: Rule RETRIEVE WORD.

The first rule (Fig. 3) attends the currently considered word. The second rule retrieves a word from the declarative memory. The third and the fourth rule (Fig. 5 and Fig. 6) shift attention to a new word in the same line and to a new word on a new line respectively. The first rule mimics the left-to-right reading due to the interplay of two requirements: (i) it is required that the new word should have the lowest x-value on the same line as the current word, (ii) at the same time, it is required that the word should not have been attended previously (by setting :attended as false). This leaves the closest word to the right as the only candidate. The jump to the left-most word in the closest lower line is achieved in a parallel way in the second rule.

One thing to notice in both rules is the value LASTWORD. This value is not specified here further, but in the actual model it would carry the position of the rightmost words on the screen, allowing the ACT-R model to shift to a new line only after the reader got to the end of the line.

As is standard, it was assumed that every rule needs 50 ms to fire.

Subsymbolic part The subsymbolic part of the ACT-R cognitive architecture is used to match human performance. Basic ACT-R reader will model eye fixations of GECO as the function of word length, frequency of the word and word

```
=g >
state                 shift
=retrieval >
cat                   =x
=visual_location >
screen_y              =ypos
screen_x              −LASTWORD
==>
=g >
state                 start
+visual_location >
:attended             False
screen_x              lowest
screen_y              =ypos
−retrieval >
```

Figure 5: Rule MOVE ATTENTION IN LINE.

```
=g >
state                 shift
=retrieval >
cat                   =x
=visual_location >
screen_y              =ypos
screen_x              LASTWORD
==>
=g >
state                 start
+visual_location >
:attended             False
screen_x              lowest
screen_y              closest
−retrieval >
```

Figure 6: Rule MOVE ATTENTION TO A NEW LINE.

position. For this reason, only two parts of the cognitive architecture will be relevant: vision module and the module of declarative memory. The rest of this section summarizes the relevant properties of these modules.

ACT-R can be used with various implementations of vision. Here, we will consider an ACT-R implementation of the EMMA (Eye Movements and Movement of Attention) model (Salvucci, 2001), which in turn is a generalization (and a simplification) of the E-Z Reader model (Reichle, Pollatsek, Fisher, & Rayner, 1998). While the latter model is used for reading, the goal of EMMA is to model any visual task, not just reading. Given the fact that the E-Z Reader model is one of the most successful models for eye-tracking data, it is natural to use its ACT-R application, EMMA, for the current purposes (see also (Engelmann, Vasishth, Engbert, & Kliegl, 2013) for another application in psycholinguistics).

Following E-Z Reader, EMMA disassociates eye focus and attention: the two processes are related but not identical.

A shift of attention to a visual object triggers (i) an immediate attempt to encode the object as an internal representation, and (ii) eye movement.

The encoding takes the time shown in Eq 1.[1]

$$T_{enc} = K \cdot D \cdot e^{kd} \tag{1}$$

In the equation, d is the distance between the current focal point of the eyes and the object to be encoded measured in degrees of visual angle (in other words, d is the eccentricity of the object relative to the current eye position), k is a free parameter, scaling the effect of distance; D is a time parameter of the object to be focused that will affect visual encoding, and K is a free parameter, scaling the encoding time itself.

In (Salvucci, 2001), it is assumed that D is a function of the (normalized) frequency of the object, $D = -log(\text{Freq})$. This assumption is present to capture the fact that high-frequent objects (words, numbers) tend to be focused shorter and skipped more often than low-frequent objects. The same effect is encoded in the E-Z reader, in which encoding time is scaled by the frequency of the object.

There is a less stipulative way to capture the effect of frequency in Basic ACT-R Reader. Objects (words) have to be retrieved from declarative memory during reading and the retrieval itself is sensitive to frequency effects. The way our symbolic system is set up will then derive the observed role of frequency on fixations and skipping indirectly and by a mechanism that is needed anyway, lexical retrieval, as we will see below. This frees Eq 1 from an extra stipulated parameter, frequency of objects. Instead of frequency, we can therefore consider other properties relevant for visual encoding. As is well-established, the length of words affects fixations and it is natural to assume that such a property would play a role when encoding an object (but not during lexical retrieval). I will assume that D is equivalent to the number of characters of a word, see Eq 2.

$$D = \text{NChar}(\text{Word}) \tag{2}$$

The time needed for eyes to move to a new object is split into two sub-processes in EMMA: preparation and execution. The preparation requires 135 ms. The execution, which follows the preparation, requires 70 ms + 2 ms for every degree of visual angle between the current eye position and the targeted visual object.[2] At the end of the execution eyes focus on the new position. If a new command to shift an attention yet again is issued during the preparation phase, the old eye movement is discarded and a new one takes place. This situation could be used to model word skipping. For more details on the interplay between attention shift and eye movements, see (Salvucci, 2001).

[1] The equation captures the time needed to encode an object if we do not assume any noise in the vision module. Otherwise, the encoding of an object is modeled using a gamma distribution with the mean T_{enc} and sd $\frac{T_{enc}}{3}$.

[2] If eye movement is assumed to be noisy, both measures are means of a gamma distribution, see the previous footnote.

The second part of the subsymbolic system important for us concerns lexical retrieval.

Simplifying somewhat and focusing only on currently relevant parameters, we can say that the time needed to retrieve a word is a function of its base-level activation. In more technical terms, we will assume that the activation of a chunk i, A_i, determining retrieval latencies, is equivalent to its base-level activation, B_i (normally, chunk activation is modulated by other chunk properties, and is distributed as Logistic(B_i, s) with s being a free parameter):

$$A_i = B_i \tag{3}$$

The base activation of a chunk in ACT-R, B_i, is in Eq 4, where d is a free parameter and t_k is the time elapsed since the chunk was presented (stored in memory).

$$B_i = \log\left(\sum_{k=1}^{n} t_k^{-d}\right) \tag{4}$$

The time needed to retrieve a chunk, T_i is shown in Eq 5. f is a free parameter, scaling the effect of the (base) activation, F is a free parameter, scaling the latency itself.[3]

$$T_i = F \cdot e^{-f \cdot A_i} \tag{5}$$

Summing up, fixation times will be affected in several ways in our model:

- The frequency of words will modulate fixation times, due to Eq 5, which becomes relevant when the rule RETRIEVE WORD (Fig. 4) fires. Frequencies will affect retrieval latencies because they affect the number and moments of chunk presentations. How frequencies are related to the number and moments of chunk presentations will not be discussed here in detail due to the lack of space. See (Reitter et al., 2011) for details, which I follow in this respect.

- The length of words will modulate fixation times, due to Eq 1 and Eq 2. These equations are relevant when the rule ATTEND WORD fires, Fig. 3. Furthermore, the length of words also influence fixation times in a less direct way. Assuming that fixations always appear at the center of a word, a word of length, say, 6 letters will make the words to the left and right appear one letter further than a word of length 4 letters. Due to the fact that executing eye movement is sensitive to distance, we should see an increase of fixation times on long words and on words preceding long words.

- Words appearing at the end of line or close to the end of line should be fixated longer. This is due to the execution time of eye movement: executing eye movement to a new line should take more time than executing eye movement to a new word on the same line.

[3] In ACT-R literature, f is not always mentioned or used. However, see (Anderson & Lebiere, 1998). The parameter will be important for our purposes.

Modeling reading

Eye-tracking reading measures are commonly split into several subtypes. The three most important ones are listed below:

- **gaze duration**: the sum of the time of all the first-pass fixations (in ms) made on a word until the point of fixation leaves the word

- **total reading time**: the sum of the time of all the fixations made on a word

- **re-reading time**: the difference between total reading time and gaze duration

The paper aims to model the effect of frequency and word properties (position, length). Such properties are standardly associated with first-pass measures. This is in fact directly encoded in E-Z Reader in which (modeled) gaze durations are functions of such factors, while re-reading measures less so (see, e.g., (Staub, 2011) for discussion and empirical evidence). Following this insight, I will focus on modeling gaze durations.

The GECO corpus stores the information about the position of each word on the screen. This enables us to fully reconstruct what each participant saw. Using this information, I re-created the reading materials of GECO.[4] I let Basic ACT-R Reader run and recorded its fixation times on every word (the value was 0 if a word was skipped). On one third of the materials, Basic ACT-R reader was run in order to find good estimates for some of its free parameters (more on this below). On one half of the materials, the model with the found parameters was studied. (The last sixth of materials was left out for possible future model comparisons.)

In the previous section, we saw five free parameters. Of these, only three were estimated: k, see Eq 1, f, see Eq 5, and F, see Eq 5. I did not model K since it would strongly correlate with F and the latter parameter might be sufficient, at least at this point (frequencies correlate with length in the data set, $r = -0.37, p < .001$). The d parameter (Eq 4) was not estimated either. Rather, its default value was used (0.5) since that is the standard and extremely common practice in ACT-R research.

As was mentioned in the introduction, parameter estimation is often done by hand in ACT-R. However, that is almost impossible to do with the amount of data that we analyze here, especially if we consider more than one parameter, as is the case here. Rather than manually finding parameter values, they were estimated using Bayesian inference and MCMC procedures. I used the Python implementation of ACT-R called PYACTR (see https://github.com/jakdot/pyactr).

[4] The materials were also cleaned and prepared for modeling. Two most important changes: frequencies from the British National Corpus based on (Leech, Rayson, et al., 2014) were added; some of the sentences had two words recorded as one if they were separated by three dots (. . .) – such sentences were excluded for two reasons. First, they would complicate the ACT-R model. Second, GECO only reports one reading measure for them and it is not clear how fixations are distributed across the two words.

(This Python implementation yields the same reaction time values for the considered parameters as the canonical implementation in Lisp.) The parameter estimation was done using the Python package for Bayesian modeling PYMC3. The Bayesian model was specified as in Eq 6. GD is the dependent variable gaze duration (in ms), Basic ACT-R(f, F, k) is a deterministic function that yields gaze duration per word by supplying Basic ACT-R Reader with the values of the three free parameters and letting the ACT-R model run. HALFNORMAL is a folded normal distribution, GAMMA is a gamma distribution, UNIFORM a uniform distribution.[5]

$$\begin{aligned} f &\sim \text{HALFNORMAL}(\mu = 0, sd = 0.5) \\ F &\sim \text{GAMMA}(\alpha = 2, \beta = 6) \\ k &\sim \text{HALFNORMAL}(\mu = 0, sd = 0.7) \\ \alpha &\sim \text{UNIFORM}(0, 200) \\ \sigma &\sim \text{HALFNORMAL}(\mu = 0, sd = 10) \\ \text{GD} &\sim \text{NORMAL}(\alpha + \text{Basic ACT-R}(f, F, k), \sigma) \end{aligned} \tag{6}$$

Notice that when retrieval and time needed to encode a word is (hypothetically) at 0 Basic ACT-R(f, F, k) should correspond only to the time needed to fire the relevant production rules. However, our current production rules are oversimplifying reading (e.g., there is no role for syntax or semantics) and thus, it is likely that they underestimate this value. This is why another parameter was added, α, and its prior was set as a non-negative value, ranging between 0 and 200 ms.

The parameters were sampled using the Metropolis sampler, with 400 steps, first 30 steps discarded and values initialized at maximum a posteriori point estimates.[6] The posterior results:

$$\begin{aligned} f &- \text{mean} : 0.15; sd : 0.09 \\ F &- \text{mean} : 0.0001; sd : 0.0001 \\ k &- \text{mean} : 0.61; sd : 0.04 \\ \alpha &- \text{mean} : 27.8; sd : 0.5 \end{aligned} \tag{7}$$

Notice that the found values f, F, k differ from the default values, which are set at 1.0. However, the default values of the last two parameters are often changed (e.g., F appears to carry the values between 0.1 and 0.4 in psycholinguistics, and k is set at 0.4 in (Salvucci, 2001)). Still, such changes do not match our found values. Unfortunately, as far as I know, previous (psycholinguistic) studies did not make systematic well-documented investigations of parameter estimates, and thus, it is completely unclear whether the differences reveal any significant discrepancies or are just accidental. The current paper is a step forward in this regard. We need to investigate free parameters of cognitive architectures in a replicable, methodical and objective way, otherwise model comparisons become impossible.

[5] When estimated, the ACT-R parameters are commonly below 0.5. I tried to reflect this by selecting prior distributions whose c.d.f at 0.5 is greater than 0.5 and have positive skew.

[6] This is a small number of steps, mainly for practical reasons: the model is slow since it has to run simulations for every word of every sentence. However, the probabilistic model is simple and the found values generate good predictions.

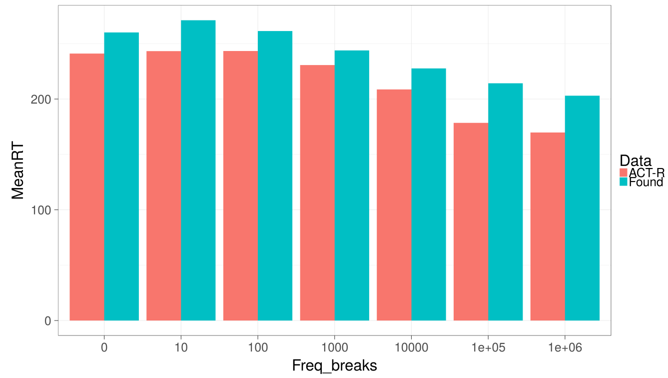

Figure 7: SimRTs and gaze durations split by word frequencies.

The mean values were plugged back into Basic ACT-R Reader. The model then simulated the reading of one half of all the sentences appearing in the GECO corpus (different sentences than the ones used in the parameter estimation). The simulated reading times (SimRTs) were used as predictors in a linear model, with mean GD (averaged across participants) as the dependent variable. The model revealed a significant effect of SimRTs ($\beta = 1.08, t = 470, p < .001$). Notice that the slope parameter β close to 1 shows that not only does Basic ACT-R Reader predict gaze durations, it does so in a way we want it to: 1 ms increase on the side of Basic ACT-R Reader corresponds to approximately 1 ms increase in actual gaze duration. The validity of the model can be also seen in Fig. 7, which plots RTs in seven frequency bands: from 0 to 10 occurrences in the BNC, from 10 to 100 etc. In each band, the red (left) bar shows mean fixation times as simulated by Basic ACT-R Reader. The right (blue) bar shows actual mean fixation times. The ACT-R model underestimates (roughly by 20 ms, which corresponds to the α estimate above) but it linearly decreases across frequency bands, closely copying the actual data. This is an encouraging finding given that the parameters were not estimated on this set of data. Fig. 8 shows that the model simulates the effect of word length well, even though it underestimates very short words, and overestimates very long words.

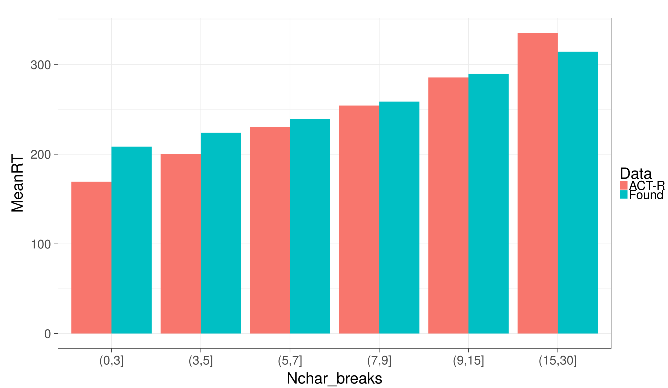

Figure 8: SimRTs and gaze durations split by word length.

An interesting question is whether the estimates of the model can be independently validated, using the same technique as above. For this reason, I used ACT-R to model a different psycholinguistic task, a lexical decision task of (Murray & Forster, 2004) (their Experiment 1). In the task, the ACT-R model (and humans) fixated the center of the screen. At that position a sequence of 5-7 letters appeared. The model (or human) then had to decide whether the sequence is an actual English word and press the corresponding key. The only manipulation relevant in the modeled experiment was that of the frequency of the appearing word. Thus, only two parameters were estimated using the data: f and F.

It is known that ACT-R is good at modeling the role of frequency in lexical decision tasks (cf. (Anderson, 1982), (Anderson, Fincham, & Douglass, 1999), (Murray & Forster, 2004)). Thus, estimates found this way might significantly strengthen our previous findings. Interestingly, f was estimated at 0.14 (sd : 0.01), thus being very close to the previously found estimate. F, in contrast, was estimated at 0.13. The difference from the previously estimated F is large, see Eq 7. In other words, while the estimated f might be close to its real value, the value of F fluctuates too wildly to be taken seriously. It remains to be seen whether it might help to model more parameters, add more information to the models or modify some other properties of the models.

Conclusion

ACT-R has been successfully used in psycholinguistics to model processing data. In this paper, I showed how it could be further expanded to model eye-tracking corpus data. The resulting model had a good fit to the corpus data, at least in the considered (low-level) processes.

Furthermore, I showed that free parameters could and should be estimated using the well-established methods in other fields, rather than by a manual search through parameter space. The latter option is impossible to use once we hit the amount of data considered here. The latter option also makes it hard, if not impossible, to compare parameters across different models since manual search is subjective and usually not well documented in research papers.

The resulting ACT-R model is a step in the direction of using ACT-R to simulate not just results of individual processing experiments, but diverse and rich corpus data. The model could be expanded to capture higher level processes (e.g., syntactic parsing). However, that is beyond the scope of this paper.

Acknowledgments

I thank SURFsara (www.surfsara.nl) for the support in using the Lisa Compute Cluster. The research presented in this paper was supported by the NWO VENI grant 275-80-005.

References

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological review*, *89*(4), 369.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford University Press.

Anderson, J. R., Bothell, D., & Byrne, M. D. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036–1060.

Anderson, J. R., Fincham, J. M., & Douglass, S. (1999). Practice and retention: a unifying analysis. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *25*(5), 1120–1136.

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

Cop, U., Dirix, N., Drieghe, D., & Duyck, W. (2016). Presenting GECO: An eyetracking corpus of monolingual and bilingual sentence reading. *Behavior research methods*, 1–14.

Engelmann, F., Vasishth, S., Engbert, R., & Kliegl, R. (2013). A framework for modeling the interaction of syntactic processing and eye movement control. *Topics in cognitive science*, *5*(3), 452–474.

Leech, G., Rayson, P., et al. (2014). *Word frequencies in written and spoken english: Based on the british national corpus*. Routledge.

Lewis, R., & Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, *29*, 1-45.

Murray, W. S., & Forster, K. I. (2004). Serial mechanisms in lexical access: the rank hypothesis. *Psychological Review*, *111*(3), 721.

Newell, A. (1973). Production systems: Models of control structures. In W. Chase et al. (Eds.), *Visual information processing* (pp. 463–526). New York: Academic Press.

Newell, A. (1990). *Unified theories of cognition*. Harvard University Press.

Reichle, E. D., Pollatsek, A., Fisher, D. L., & Rayner, K. (1998). Toward a model of eye movement control in reading. *Psychological review*, *105*(1), 125.

Reitter, D., Keller, F., & Moore, J. D. (2011). A computational cognitive model of syntactic priming. *Cognitive science*, *35*(4), 587–637.

Salvucci, D. D. (2001). An integrated model of eye movements and visual encoding. *Cognitive Systems Research*, *1*(4), 201–220.

Staub, A. (2011). Word recognition and syntactic attachment in reading: Evidence for a staged architecture. *Journal of Experimental Psychology: General*, *140*, 407–433.

Taatgen, N. A., & Anderson, J. R. (2002). Why do children learn to say "broke"? a model of learning the past tense without feedback. *Cognition*, *86*(2), 123–155.

Vasishth, S., Bruüssow, S., Lewis, R. L., & Drenhaus, H. (2008). Processing polarity: How the ungrammatical intrudes on the grammatical. *Cognitive Science*, *32*, 685–712.