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

Author: Jakub Dotlačil

Mailing address:
Jakub Dotlačil
ILLC
Universiteit van Amsterdam
P.O. Box 94242
1090 GE AMSTERDAM
The Netherlands

e-mail: j.dotlacil@gmail.com

Department & Institution:
Institute for Logic, Language and Computation, University of Amsterdam

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

November 9, 2017

Abstract

Cognitive architectures have often been applied to data from individual experiments. In this paper, I develop an ACT-R reader that can 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. Unlike previous related works (most prominently, F., Engelmann, Vasishth, S., Engbert, R., & Kliegl, R, 2013), the model achieves the fit by estimating free parameters of ACT-R using Bayesian estimation and Markov-Chain Monte Carlo (MCMC) techniques, rather than by relying on the mix of manual selection + default values. The method used in the paper is generalizable beyond this particular model and data set and could be used on other ACT-R models.

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

1 Introduction

Language represents a detailed and richly structured part of cognition that is integrated with many general cognitive sub-systems, e.g., learning, perception, memory, planning. For this reason, it is not surprising that cognitive architectures, which integrate various findings of cognitive sciences into one theory, should attempt to model language knowledge. Of various architectures, Soar (see (Newell, 1990)) and ACT-R (Adaptive Control of Thought–Rational) (see (Anderson, Bothell, & Byrne, 2004) for an introduction) have probably been applied most often to this task. For example, Soar has been used to study syntactic parsing in (Hale, 2014), while ACT-R has been implemented in research on parsing, memory retrieval, syntactic priming or the reanalysis of syntactic structures (Lewis & Vasishth, 2005; Vasishth, Bruüssow, Lewis, & Drenhaus, 2008; Reitter, Keller, & Moore, 2011, among others).

Focusing on language has one advantage compared to several other domains of cognitive research, namely, linguistics provides modelers with a very rich pool of data that are instantly available in language corpora. Whilst a common practice in studies of integrated cognition is to model selected experimental results and this position is insightful (see (Anderson & Lebiere, 1998), (Anderson, 2007), for many examples in ACT-R), corpora can be an important source in modeling. First, they provide data that are naturally occurring, rather than carefully constructed, and therefore, modeling such data increases the ecological validity of underlying theories. Second, corpora provide a vast amount of data points (compared to findings in individual experiments). This is not useful per se, but it can be useful for cognitive architectures which specify only general constraints, leaving a lot of freedom to modelers. Narrowing down a space of possible models is hard to do with small data sets, making corpora a very helpful tool.

Several previous applications of cognitive architectures made use of corpus data. For example, (Taatgen & Anderson, 2002) modeled children's spontaneous speech, (Engelmann, Vasishth, Engbert, & Kliegl, 2013) evaluated a syntactic retrieval model of (Lewis & Vasishth, 2005) in the Potsdam Sentence Corpus (Kliegl, Grabner, Rolfs, & Engbert, 2004), (Hale, 2014) studied production compilation in the Dundee eye-tracking corpus (Kennedy & Pynte, 2005). One challenge that such works face is specification of free parameters in models, that is, parameters that can be manipulated to improve the fit of the underlying theoretical concepts to the human performance.

There are two common practices to link concepts to performance via free parameters. One approach is to find values for parameters by hand and to assess found values by measures of goodness of fit (e.g., R^2 or $RMSE$, the root mean squared error). This method is used, for example, in (Engelmann et al., 2013) and probably also in (Taatgen & Anderson, 2002). But this becomes very tedious when fitting corpus data, which requires long runs of simulation and as a consequence, only a narrow space of possible parameter values might be explored. Furthermore, unless it is explicitly stated what values were explored, the method is hard to replicate. The most radical version of this method is the second approach, which leaves parameters in their default values (so-called zero-parameter fit). While the method is very easy to replicate, as well as very restrictive, it under-explores possible predictions of the model (see (Roberts & Pashler, 2000)). Furthermore, not every parameter might be associated with a default value, or more values might be commonly used.

In this paper, I diverge from such approaches and show how parameters of a cognitive architecture can be fitted by embedding the architecture in a Bayesian model and using Markov-Chain Monte Carlo (MCMC) methods to sample target distributions. For the task, I use ACT-R since it is arguably the most dominant cognitive architecture used in (psycho)linguistics.

Unlike the manual selection of the parameters, this approach is objective, easy to document and to replicate since the considered parameter space becomes part of the specifications of the Bayesian model. The method also scales up to large data sets straightforwardly. It is also shown that using MCMC methods

makes it easy to compare parameters across 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.

A very similar approach to the one considered here has been taken in (Weaver, 2008), which argues for Bayesian estimation of ACT-R parameters and ACT-R models in general (the latter issue is not considered here). (Weaver, 2008) focuses on statistical underpinnings and provides a proof of concept of this approach, applying the method to only minimal models and simulated data. The current paper shows that ACT-R can benefit from Bayesian modeling even when we scale up and consider complex data and much more complicated and realistic models (for example, we are going to use an asynchronous model, in which, as is very common ACT-R, two or more processes run in parallel, while (Weaver, 2008) studies only synchronous ones). But much more importantly, the goal of the paper is practical. I believe that by following the discussion of one particular model described in the paper, cognitive modelers should be able to apply Bayesian estimation to parameter specification in a model of their liking. It does not require more than the basic knowledge of ACT-R, Python and Bayesian statistics.

2 Modeled data

The paper presents a model of (a subset of) reading measures of the Ghent Eye-Tracking Corpus, GECO (Cop, Dirix, Drieghe, & Duyck, 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. Given its size (54,364 words, 5,300 sentences) it belongs among the largest eye-tracking corpora. 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).

3 Basic ACT-R reader[1]

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

[1]The model is available at https://www.github.com/jakdot/published_models. The model is written in Python and requires pyactr for ACT-R modeling, and MPI for parallel computation.

AFRAID [ISA: word form: afraid category: adjective]

Figure 1: Example of the chunk AFRAID.

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 a line, the reader shifts its visual focus to the beginning of the next line and proceeds in reading. After the last word of the sentence, the first word of the next sentence is 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 useful in modeling eye-tracking corpus data. Once the model is in place, we can move to more complex cases.

3.1 Symbolic part

As is well-known, ACT-R subsumes two types of knowledge: declarative knowledge and procedural knowledge (see (Newell, 1990) on the difference). A common understanding is that the declarative knowledge represents our knowledge of facts, the procedural knowledge is the knowledge that we display in our behavior (see (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), which are nothing more than conditional actions (do B if in state A).

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 Figure 1.

The procedural knowledge consists only of a handful of rules: *attend word*, *retrieve word*, *move attention in line*, *move attention to a new line*.

The first rule (*attend word*) attends the currently considered word, i.e., it moves visual attention to the position of the word. The second rule (*retrieve word*) retrieves an attended word from the declarative memory. The third and the fourth rule (*move attention in line*, *move attention to a new line*) shift attention to a new word in the same line and to a new word on a new line respectively. Slightly more abstractly, the rules form a loop, shown in Figure 2.

The rule *move attention in line* 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,

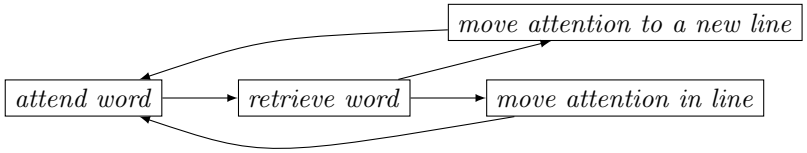

Figure 2: Sequence of production rules

it is required that the word should not have been attended previously.[2] This leaves the closest word to the right as the only candidate. The jump to the leftmost word in the closest lower line is achieved in a parallel way in the rule *move attention to a new line*.

As is standard in ACT-R, it is assumed that every rule needs 50 ms to fire.[3]

3.2 Subsymbolic part

3.2.1 Visual encoding

Basic ACT-R reader models eye fixations of GECO as the function of word length, frequency of the word and word position. For this reason, only two subsymbolic parts of the cognitive architecture are 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 reading data, it is natural to use its ACT-R application, EMMA, for the current purposes (see also (Engelmann et al., 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 Equation 1.[4]

[2]This second condition is simulated in ACT-R by letting the model attach pointers to attended objects. The number of pointers ACT-R can remember can be limited by specifying finsts (fingers of instantiation). As it is, the model would need to use at least as many finsts as there could be words on one line for this rule to work. We ignored the role of finsts.

[3]For visual encoding, ACT-R models sometimes used 10 ms firing rules, as in (Salvucci, 2001). But this divergion from the standard assumption does not seem necessary: (Engelmann et al., 2013) fit their reading model using the default ACT-R time for rule firing.

[4]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 $\frac{T_{enc}}{3}$.

$$T_{enc} = K \cdot D \cdot e^{kd} \quad (1)$$

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 can capture the fact that high-frequent words tend to be focused shorter and skipped more often than low-frequent words (see (Engelmann et al., 2013) for details). 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 reading using ACT-R. Encoding of a word as a visual object is not an end goal in itself. Rather, it is just the first step, after which a chunk with the same properties (i.e., the same form in case of words) is retrieved from declarative memory for interpretational purposes. Since the retrieval itself is sensitive to frequency effects, it is more conservative to derive the observed role of frequency on fixations and skipping indirectly and by a mechanism that is needed anyway. In our case, frequency effects will arise due to lexical retrieval, as we will see below. Instead of word frequency, we could consider other, purely visual, properties for D. We will take only one visual aspect into account. 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 attending visual properties of an object. I will assume that D is equivalent to the number of characters of a word, see Equation 2.

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

The time needed for eyes to move to a new object is split into two subprocesses 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.[5] 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).

3.2.2 Lexical retrieval

The second part of the subsymbolic system in the ACT-R model concerns lexical retrieval.

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

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 Equation 4, where d is a free parameter and t_k is the time elapsed since the chunk was presented (stored in memory). If a chunk is presented just once, Equation 4 simplifies into Equation 5.

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

$$B_i = \log\left(t^{-d}\right) \tag{5}$$

I will now indicate how frequency can be translated into activation (see also (Reitter et al., 2011)).

Consider a 15-year old speaker. How can we estimate the time points at which a word was used in language interactions that the speaker participated in?

We know the lifetime of the speaker (15 years), so if we know the total number of words an average 15-year old speaker has been exposed to, we can easily calculate how many times a particular word was used on average based on their frequency. Once we find out how many times a word with a specific frequency was presented to our speaker during their lifetime, we can then present the word at linearly spaced intervals during the life span of the speaker. A good approximation of the number of words a speaker is exposed to per year can be found in (Hart & Risley, 1995). Based on recordings of 42 families, Hart and Risley estimate that children comprehend between 10 million to 35 million words a year, depending to a large extent on the social class of the family, and this amount increases linearly with age. According to the study, a 15-year old has been exposed to anywhere between 50 and 175 million words total. For simplicity, the model will work with the mean of 112.5 million words as the total amount of words a 15-year old speaker has been exposed to. This is a conservative estimate as it ignores production and the linguistic exposure associated with mass media. But under- and overestimates can be corrected by scaling parameters, so this should not be of concern.

The time needed to retrieve a chunk, T_i is shown in Equation 6. The variable f is a free parameter, scaling the effect of the (base) activation, F is a free parameter, scaling the latency itself.[6]

[6]In ACT-R literature, f is not always mentioned or used. See (Anderson & Lebiere, 1998) for discussion and (West, Pyke, Rutledge-Taylor, & Lang, 2010) for the application of f to model retrieval latencies. The parameter will be important for our purposes.

$$T_i = F \cdot e^{-f \cdot A_i} \tag{6}$$

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

- The frequency of words will modulate fixation times, due to Equation 6, which becomes relevant when the rule *retrieve word* fires. Frequencies will affect retrieval latencies because they affect the number and moments of chunk presentations, that is, they are responsible for the base-level activation of chunks.
- The length of words will modulate fixation times, due to Equation 1 and Equation 2. These equations are relevant when the rule *attend word* fires. Furthermore, the length of words also influences 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.
- Words appearing at the end of line should be fixated longer. Similarly, even words appearing close to the end of line might be fixated longer in case the following words on the same line are skipped. This is due to the execution time of eye movement: since going from the rightmost word in one line to the leftmost word in another line requires crossing a longer distance than moving from word to word within one line, the execution time of eye movement, dependent on distance, should increase.

4 Modeling reading

Eye-tracking reading measures are commonly split into several subtypes. The three most important ones are listed below, see also Figure 3:

- **gaze duration**: the sum of the time of all the first-pass fixations 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., (Kliegl et al., 2004), (Staub, 2011) for discussion and empirical evidence). Following this insight, I will focus on modeling gaze durations.

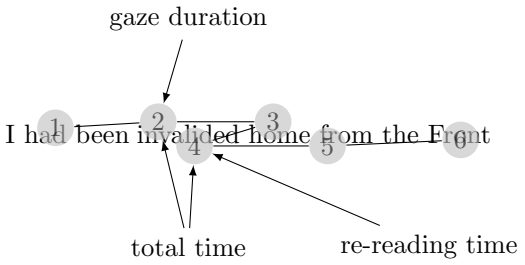

Figure 3: Visualization of eye-tracking reading measures on the word *invalided*. Gray circles represent fixations, lines represent saccades. Gaze duration = fixation 2; total time = fixation 2 + fixation 4; re-reading time = fixation 4

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.[7] I let Basic ACT-R Reader run and recorded its fixation times (the value was 0 if a word was skipped). On one fourth 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 another fourth of the materials, the model with the found parameters was studied. The rest of data was left out for future investigations.

Even though the model itself read every word, fixation times at the first and the last word of any sentence were not recorded. Ignoring first and last words is a common practice in models of eye fixations, (Reichle et al., 1998; Salvucci, 2001; Engelmann et al., 2013).

In the previous section, we saw that the model operates with five free parameters. Of these, only three were estimated: k, see Equation 1, f, see Equation 6, and F, see Equation 6. 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 (Equation 4) was not estimated either. Rather, its default value was used (0.5).

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 is analyzed here, especially if, as is the case here, we consider more than one parameter. We might want to abstract away from symbolic parts of the model, estimating only parameters by relating gaze duration directly to equations 1 to 6. But how could that be done? Consider, for example, an attempt to

[7]The materials were also cleaned and prepared for modeling. Two most important changes: (i) frequencies from the British National Corpus based on (Leech, Rayson, et al., 2014) were added (words present in the eye-tracking corpus but missing from the BNC received frequency of 1, so they could be recalled), and (ii) sentences that had two words separated by three dots (. . .) were excluded. The exclusion was done mainly because sentences with dots would complicate the ACT-R model, as it is not clear whether to treat such words as one object or two objects and because GECO does not report the exact position of the word that appears after the three dots.

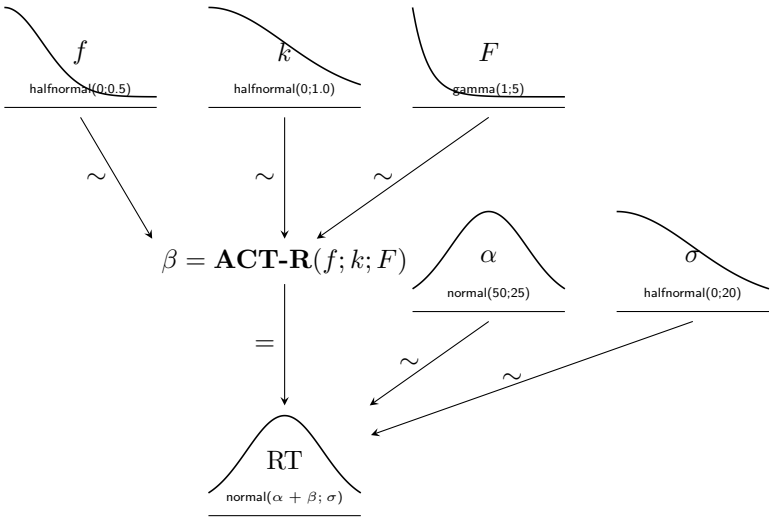

Figure 4: Bayesian model for parameter estimation

estimate just F and f in Equation 4 and Equation 6 as parameters that modulate retrieval latencies. A problem is that it is not clear how retrieval latencies relate to eye fixations, which is the only dependent measure we have. Of course, we could assume that retrieval latencies are linearly related to fixations, but that would go against the EMMA model we considered so far. In that model, the relation is monotonic but non-linear. For example, retrieval latencies might be masked if they happen during the preparation phase and the execution phase of eye movements, or a skipped word would increase reading latencies on some preceding word. Such interactions make it very hard to simplify any parts of the model for the purposes of parameter estimation.

In this paper, I follow a different route: I estimate parameters by embedding all the relevant parts of the ACT-R model in a Bayesian model. For that, I used the Python implementation of ACT-R PYACTR (see https://github.com/jakdot/pyactr), which 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 probabilistic programming PYMC3. The Bayesian model was specified as in Figure 4. RT is the dependent variable gaze duration (in ms), Basic ACT-R$(f; F; k)$ is a 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 to simulate β. *halfnormal* is a folded normal distribution, *gamma* is a gamma distribution. The graph in the figure follows the conventions of (Kruschke, 2011), in which dependent variables appear at the bottom of the figure and arrows signal dependency. Every variable appears with the graphical representation of its prior probability distribution. The tilde represents non-deterministic dependency, while '=' is deterministic. Note that

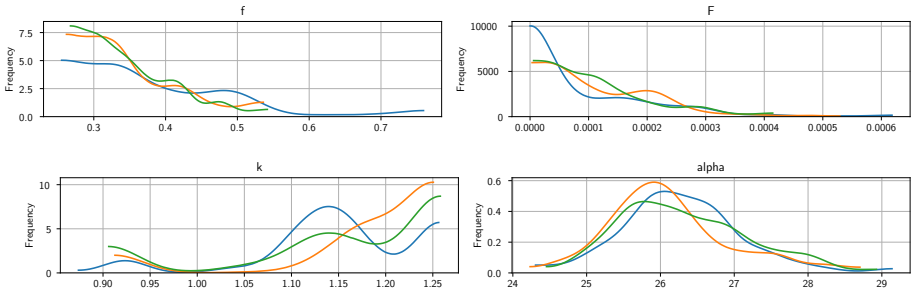

Figure 5: Frequency of drawn samples from 3 chains after burn-in for f, F, k, α

Basic ACT-R($f; F; k$) is deterministic. This is not a necessary feature, for example, we could expand the model by assuming that chunk activation or eye movements are noisy. However, that would complicate the model, since additional parameters would have to be estimated.

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 over-simplifying 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 distribution that is more likely to be positive (mean=50 ms) even though nothing much is known about it (SD=25 ms). Notice that this parameter is a constant adjustment (it can increase or decrease RTs for all words), it does not interact with other parameters and therefore, it is not used to predict what we care about: the effect of frequency and visual properties on eye fixations.

A brief note about the prior distributions for the ACT-R parameters follows. The parameters f, k and F in most previous studies carry values between 0 and 1. This is reflected in priors, i.e., cumulative distribution functions assign 96% probability and 68% probability to f and k at 1. Furthermore, of the three parameters, F is usually set at the lowest values: it has been set at the value of 0.14 for several processing studies (see (Lewis & Vasishth, 2005)) even though higher values were also considered ((Vasishth et al., 2008) use the value of 0.46). Given this, the used prior distribution for F with mean of 0.2 and the most density assigned to values between 0 and 0.5 is arguably appropriate.

The parameters were sampled using the Metropolis algorithm, with 1000 steps, burn-in 400 steps and 3 chains initialized at random starting points.[8]

[8] A few notes on these choices. Unlike more recent samplers that are implemented in PYMC3 (e.g., No-U-Turn Sampler), the Metropolis algorithm is compatible with user-defined functions/probability distributions, a feature that is exploited here, hence its use. Sampling with 1000 steps is short but it was done for practical reasons: even with 150 parallel processes per chain, one iteration took around 5 minutes. Furthermore, the statistical model is not complex and therefore, this number of steps is likely to be sufficient. To make sure that it was sufficient, the convergence of chains was further studied, see the main text.

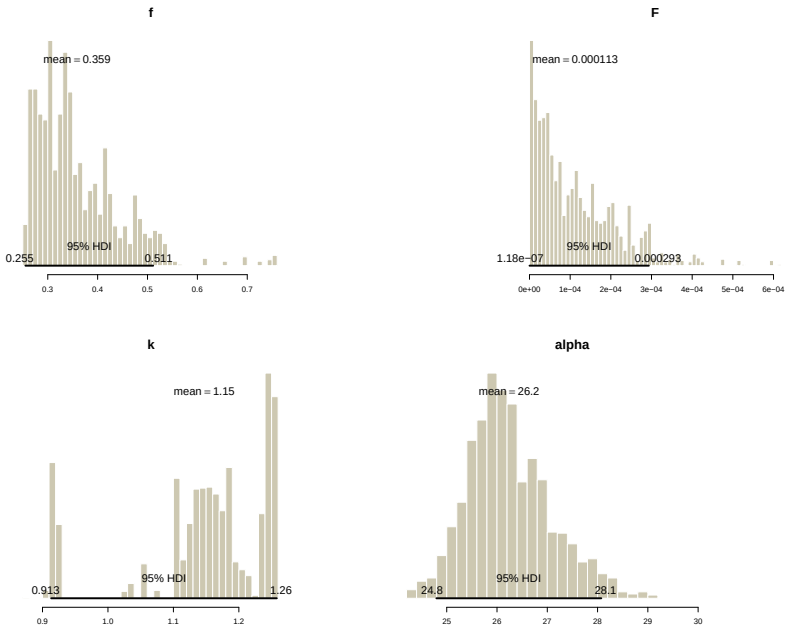

Figure 6: Posterior distributions

The chains converged, as can be seen from Rhat values ($\hat{R} < 1.1$ for all parameters)[9] and visual inspection of sampled frequency in Figure 5. The posterior results, including means and highest density intervals, are summarized in Figure 6. Bayesian estimation yields distributions, rather than point estimates (see (Weaver, 2008) why this is better).

To further evaluate the model, I plugged the mean values of f, F and k back into Basic ACT-R Reader and let the model simulate the reading of one fourth of GECO sentences, different from the sentences that were used for parameter estimation. The simulated reading times (SimRTs) were used as predictors in a linear model, with mean gaze duration (averaged across participants; words skipped by all participants treated as missing values) as the dependent variable. The model revealed a significant effect of SimRTs ($\beta = 0.97, t = 373, p < .001$). 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 Figure 7, which plots the found and simulated data as a function of frequency and length. In each band, the red (left) bar shows mean fixation times as simulated by Basic ACT-R Reader. The right (blue) bar shows actual

[9]See (Gelman & Rubin, 1992) for a definition of $\hat{R}$. In short, $\hat{R}$ close to 1 signals that simulated observations of variables approximate target distributions well.

Frequency	No. of tokens
(0, 10]	38
(10, 100]	156
(100, 1000]	468
(1000, 10,000]	1,121
(10,000, 100,000]	1,864
(100,000, 1 million]	3,057
(1 million ...)	1,597

Nchar	No. of tokens
(0, 3]	3,983
(3, 5]	2,472
(5, 7]	1,114
(7, 9]	525
(9, 11]	150
(11, ...)	57

Table 1: Summary of no. of tokens in the evaluated data set by frequency and length

mean fixation times. The match between predicted RTs and actual data is encouraging given that the parameters were not estimated on this set of data. Table 1 summarizes the number of tokens in the evaluated data set as a function of frequency or length.

Apart from the effect of frequency and length, Basic ACT-R Reader also predicts increased RTs at the end of every line in the studied data set ($\beta = 45$ ms, $p < .001$). However, an opposite effect was found in the actual data, i.e., in gaze duration ($\beta = -26$ ms, $p < .001$). This suggests that some modifications of EMMA might be needed: while execution phase for eye movements should normally increase with visual distance, this increase is unwarranted in moving eyes to a new line, probably because the position of the first word on the next line is highly predictable.

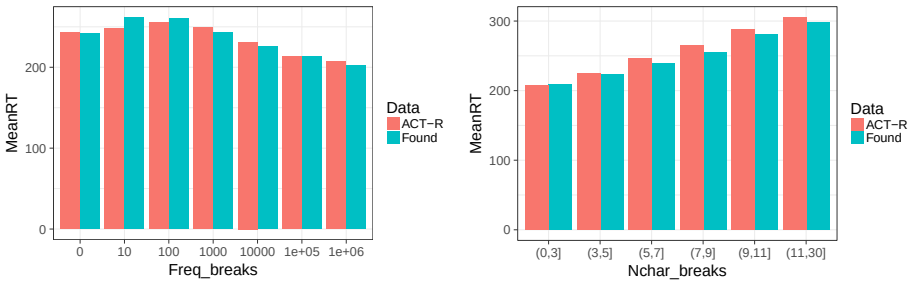

Figure 7: SimRTs and gaze durations split by word frequencies.

Readers familiar with ACT-R probably noticed that the estimated parameters differ from the default values, which are set at 1.0 (for f and F) and at 0.4 (for k), and the commonly considered values for F range between 0.1 and 0.4. Such values do not match what we found.

Sometimes, the fact that the model uses default values is considered as its strength and modifying values as its weakness (see, e.g., p. 184 of (Anderson & Lebiere, 1998), p. 707 of (Vasishth et al., 2008)), since adhering to zero-parameter fit constrains the theory. However, from the perspective of transparency, it is less obvious. Value modification is directly interpretable: it sig-

nals that the default value resulted in worse fit to data (why would the modeler change it otherwise?) and modifications improved the fit. In contrast to that, the use of default values cannot be interpreted in just one way: it could mean that (i) the default values yield the best fit of the model, (ii) the model fits sub-optimally but the modeler did not consider changing the values, (iii) changing defaults did not matter, e.g., because parameters are correlated and changing one parameter could be "fixed" by changing another parameter correspondingly. Given this, default values might be overused and have little theoretical significance (i.e., such values might be used because of tradition/ease, rather than because they are the best fit so far). Ideally, we should show with the use of data-driven methods, as the one considered here, that various models converge on similar distributions for free parameters. That would form a strong support for particular shapes and ranges of values, one that other models should take into account. But since a data-driven model fitting strategy is uncommon, it is hard to compare the posterior values of f, F and k to other models, and thus, it is unclear whether the differences between our findings and, say, default values reveal any significant discrepancies or are just accidental. The current paper is a step forward in this regard.

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.

It is known that ACT-R is good at modeling the role of frequency in lexical decision tasks (see (Anderson, 1982), (Anderson, Fincham, & Douglass, 1999), (Murray & Forster, 2004)). Thus, estimates found this way might strengthen our previous findings. Interestingly, f was estimated at 0.28 (HDI: $[0.06-0.48]$), thus being close to the estimate found here. F, in contrast, was estimated at 0.45. The difference from the estimated F is large. 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.

5 Conclusion

ACT-R has been successfully used in psycholinguistics to model processing data. In this paper, I showed how it could be 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 how free parameters could be estimated using the well-established methods in other fields, rather than by a manual search through parameter space. The latter option is hard, if not 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.

6 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.

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, 457–472.

Hale, J. T. (2014). *Automaton theories of human sentence comprehension.* Stanford: CSLI Publications.

Hart, B., & Risley, T. R. (1995). *Meaningful differences in the everyday experience of young american children.* Baltimore: Paul H Brookes Publishing.

Kennedy, A., & Pynte, J. (2005). Parafoveal-on-foveal effects in normal reading. *Vision research*, *45*(2), 153–168.

Kliegl, R., Grabner, E., Rolfs, M., & Engbert, R. (2004). Length, frequency, and predictability effects of words on eye movements in reading. *European Journal of Cognitive Psychology*, *16*(1-2), 262–284.

Kruschke, J. K. (2011). *Doing Bayesian data analysis: A tutorial with R and BUGS*. Academic Press/Elsevier.

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.

Roberts, S., & Pashler, H. (2000). How persuasive is a good fit? a comment on theory testing. *Psychological review*, *107*(2), 358–367.

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.

Weaver, R. (2008). Parameters, predictions, and evidence in computational modeling: A statistical view informed by act–r. *Cognitive Science*, *32*(8), 1349–1375.

West, R., Pyke, A., Rutledge-Taylor, M., & Lang, H. (2010). Interference and act-r: New evidence from the fan effect. In D. D. Salvucci & G. Gunzelmann (Eds.), *Proceedings of the 10th international conference on cognitive modeling* (p. 211-216). Philadelphia, PA: Drexel University.