COGNITIVE SCIENCE
A Multidisciplinary Journal

Cognitive Science 45 (2021) e13020

ISSN: 1551-6709 online
DOI: 10.1111/cogs.13020

Parsing as a Cue-Based Retrieval Model

Jakub Dotlačil

Utrecht Institute of Linguistics, Utrecht University

Received 19 August 2020; received in revised form 9 June 2021; accepted 26 June 2021

Abstract

This paper develops a novel psycholinguistic parser and tests it against experimental and corpus reading data. The parser builds on the recent research into memory structures, which argues that memory retrieval is content-addressable and cue-based. It is shown that the theory of cue-based memory systems can be combined with transition-based parsing to produce a parser that, when combined with the cognitive architecture ACT-R, can model reading and predict online behavioral measures (reading times and regressions). The parser's modeling capacities are tested against self-paced reading experimental data (Grodner & Gibson, 2005), eye-tracking experimental data (Staub, 2011), and a self-paced reading corpus (Futrell et al., 2018).

Keywords: Computational psycholinguistics; Cue-based retrieval; Memory retrieval; ACT-R; Modeling reading data; Processing

1. Introduction

Human parsing, that is, syntactic-structure building, relies on memory in at least two ways. First, it happens often that an element might be dependent in its interpretation and/or form on some other, non-adjacent phrase, and language users need to be able to access the phrase when constructing a correct parse. For example, in (1), the noun phrase (NP) *a book* is interpreted as the object of the verb *love* and if the parser is to correctly establish the relation, it has to be able to access the NP in its memory when the verb is parsed.

It is a book that I think the American readership will love immediately. (1)

Correspondence should be sent to Jakub Dotlačil, Utrecht University, Trans 10, 3512JK Utrecht, The Netherlands. E-mail: j.dotlacil@gmail.com

There is another, even more basic role that memory plays in human parsing. Comprehenders have to rely on their memory of parsing rules when they try to understand a written or a spoken message. For example, for (1), they have to remember grammar conventions like (i) *that* can introduce a relative clause, (ii) subjects precede verbs in non-transformed structures, (iii) the object in object-relative clauses has to be recalled when one hears the main verb, etc.

In recent years, research on memory in parsing has focused on the first role of memory during comprehension. The investigation of dependencies such as *a book–love* in (1) provides a growing body of evidence that this part of human parsing can be modeled as a case of cue-based retrieval. The evidence for cue-based retrieval of dependents comes from various experimental methods, from reading (self-paced reading and eye tracking; Cunnings & Sturt, 2018; Dillon, Mishler, Sloggett, & Phillips, 2013; Kush, Lidz, & Phillips, 2015; Van Dyke, 2007) to speed-accuracy trade-off (McElree, 2000; Mcelree, Foraker, & Dyer, 2003; McElree, 2006), and was further supported by Bayesian meta-analysis of experimental data (Engelmann, Jäger, & Vasishth, 2019; Jäger, Engelmann, & Vasishth, 2017; Vasishth, Nicenboim, Engelmann, & Burchert, 2019).

The success of this research line, however, leads to a schism in the general theory of parsing. While psycholinguists currently have a detailed theory of memory structures for the processing of dependencies, the theory of how parsing rules are structured, stored, and recalled is arguably less specific. This schism is probably most apparent in computational psycholinguistic models. In models that focus on retrieval during parsing, that is, models of processing of dependencies, the role of parsing is either simplified (Dillon et al., 2013; Dubey, Keller, & Sturt, 2008; Kush et al., 2015) or is constructed in such a way that the (retrieval of a) parsing rule makes no clear and generalizable behavioral footprint (Brasoveanu & Dotlačil, 2018; Gibson, 1998; Lewis & Vasishth, 2005; Rasmussen & Schuler, 2018). In models that focus on parsing, a linking hypothesis that is responsible for connecting parsers to behavioral data is usually independent of memory assumptions. Computational psycholinguistic models of human parsing that predict behavioral measures commonly assume that other properties, for example, relative entropy of parsed structures, prefix probabilities, are relevant explanatory variables (Hale, 2001, 2003, 2011; Levy, 2008, 2011), not the same memory structures that account for dependency resolution and that are assumed in cue-based retrieval. To be sure, there are parsing models that do operate with memory and memory limitations but those assume a separate model for parsing and for the resolution of dependencies (Boston et al., 2011; Demberg & Keller, 2008).

This paper represents an attempt to connect the two strands of research in parsing by developing a cue-based retrieval system of parsing. The goal of the paper is the following:

1. To provide a data-driven parser that postulates parsing rules in memory and assumes cue-based retrieval. It will be shown that there is a class of parsers in computational linguistics that are compatible with this position.
2. To show that the parser can be embedded in a cognitive architecture, ACT-R. Because the architecture simulates human behavior, this will enable the parser to predict behavioral data.
3. To study the predictions of the parser. The predictions will be investigated on three different data sets:

- Grodner and Gibson (2005), in which parsing is intertwined with recall of dependents
- Staub (2011), in which parsing is intertwined with lexical retrieval
- Corpus data from Futrell et al. (2018)

We will see that the model can fit the experimental results very well and provide a strongly significant predictor for reading data. The paper thus provides support for a psycholinguistic parser that is built on independently established properties of human memory. Since the parser is inspired by cue-based retrieval models, it will be labeled throughout as "the cue-based model of parsing."

The structure of the paper is as follows. In Section 2, the general schema of cue-based retrieval models is presented and it is shown how the schema is implemented in the cognitive architecture ACT-R. In Section 3, a brief introduction into transition-based parsers is given. Section 4 is the core of the paper: it provides various modeling evidence for the parser. Section 5 compares the cue-based model of parsing to other related work in computational linguistics and psycholinguistics. Section 6 concludes.

2. ACT-R and cue-based retrieval

This section summarizes the main claims of the theory of cue-based retrieval and explains how cue-based retrieval is enforced in the cognitive architecture Adaptive Control of Thought-Rational (ACT-R). The latter point is crucial for the follow-up sections, since the data-driven parser, introduced in Section 3, will also be implemented in ACT-R.

2.1. Basic case of cue-based retrieval

The basic idea of the cue-based model will be presented on the four-sentence paradigm in (2) and (3). The sentences investigate the retrieval of the subject noun in subject–verb dependencies. The examples in (2) are taken from Van Dyke (2007). The examples in (3) come from Wagers, Lau, and Phillips (2009) and are based on Pearlmutter, Garnsey, and Bock (1999).

a. The worker was surprised that the resident who was living near the dangerous neighbor *wascomplaining* about the investigation.

b. The worker was surprised that the resident who was living near the dangerous warehouse *wascomplaining* about the investigation. (2)

a. The key to the cell unsurprisingly *wererusty* from many years of disuse.

b. The key to the cells unsurprisingly *wererusty* from many years of disuse. (3)

When readers parse the verb phrase *was complaining* in 2 and *were rusty* in 3, italicized in the example, they have to recall the subject for the correct interpretation of the argument structure. When searching for the correct noun in their memory, the cue-based model assumes

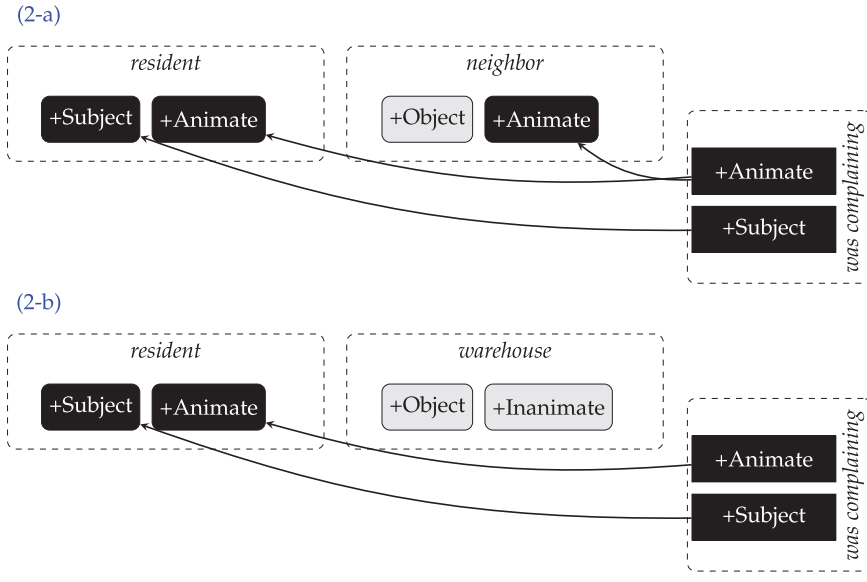


Fig. 1. Retrieval for examples (2-a) and (2-b). The retrieval is driven by features [+SUBJECT], [+ANIMATE]. The features are either matched (black rectangle) or mismatched (gray rectangle). For this illustration, we assume that only two NPs compete for recall, the target (*the resident*), or the distractor (*the neighbor*/*the warehouse*). The overload of the cue [+ANIMATE] in the top example should cause inhibitory interference according to the presented theory of retrieval.

that they can try to match searched nouns against several features. We will focus only on those features that are crucial for the predictions of the cue-based model. For our discussion of (2-a) and (2-b), the features we have to consider are [+SUBJECT] and [+ANIMATE]. The latter cue can be thought of as triggered by thematic restrictions of the verb *complaining*. For (3-a) and (3-b), the relevant features are [+SUBJECT] and [+PLURAL]. I will first discuss the predictions of the theory and then explain how they are derived in ACT-R from theoretical principles.

Let us focus on (2-a) and (2-b) for now. The features relevant for the discussion are schematically represented in Fig. 1. When we compare the two cases in Fig. 1, we see that in (2-a), the cue [+ANIMATE] is overloaded: it is matched by the subject that should be recalled, *the resident*, as well as the non-subject distractor, *the neighbor*. This cue overload should, according to the presented theory of retrieval, lead to the inhibitory interference of the distractor in (2-a) compared to (2-b). In reading times, the inhibition should manifest itself by slowdown. Such slowdown was observed for subject–verb dependencies in the studies that investigated the syntactic and semantic overload effect (see Van Dyke & Lewis, 2003; Van Dyke & McElree, 2006; Van Dyke, 2007; Jäger et al., 2017). At least one study also found the slowdown effect caused by the overload of the morphological information, number (Nicenboim, Vasishth, Engelmann, & Suckow, 2018).

The case of (3-a) and (3-b) is represented in Fig. 2. In this pair, neither sentence is grammatical since the target and the distractor result only in a partial match. Still, the theory predicts that the partial match of the distractor affects retrieval. The partial match should lead to the

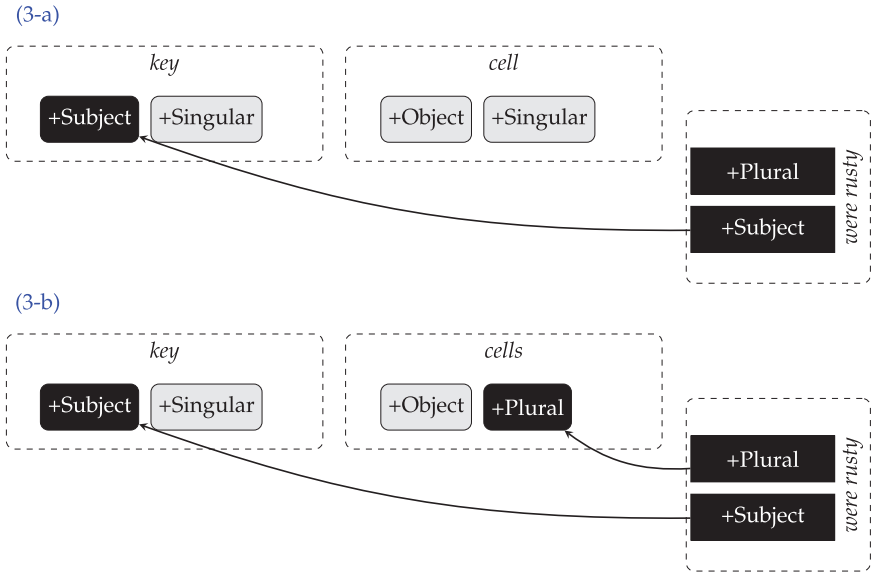


Fig. 2. Retrieval for examples c and d in 2. The retrieval is driven by features [+SUBJECT], [+SINGULAR] on the verb. The features are either matched (black rectangle) or mismatched (gray rectangle). There are two nouns that could be recalled, the target (*key*) and the distractor (*cell/cells*). Note that in this case, neither the target nor the distractor fully match. The partial match of the distractor should lead to the facilitatory interference according to the presented theory of retrieval.

facilitatory interference of the distractor in (3-b) compared to (3-a). In reading times, the facilitation should manifest itself by speedup. Such effects were observed for subject–verb number dependencies (see Dillon et al., 2013; Jäger et al., 2017; Jäger, Mertzen, Van Dyke, & Vasishth, 2020; Lago, Shalom, Sigman, Lau, & Phillips, 2015; Tucker, Idrissi, & Almeida, 2015; Villata, Tabor, & Franck, 2018; Wagers et al., 2009, among others).

2.2. *Declarative memory in ACT-R and cue-based retrieval*

There are two dominant theories in psycholinguistics that generate the just-summarized predictions for (2) and (3): ACT-R and the Direct Access model (for comparison, see Nicenboim & Vasishth, 2018; Vasishth et al., 2019). Let us see how the findings are captured in the former theory. I focus on ACT-R because it is much more encompassing and general than the Direct Access Model. ACT-R is not just a model of cue-based retrieval. It is a cognitive architecture, which can simulate the interaction of memory with execution, planning, visual perception, motor control, etc. (see Anderson & Lebiere, 1998; Anderson et al., 2004; Anderson, 2007). For this reason, it will also be suitable for the construction of the cue-based model of parsing.

ACT-R assumes two types of memory: procedural memory and declarative memory. I focus here on the latter and very briefly describe how retrieval from declarative memory works (for more details, motivation and a more beginner-friendly introduction, see Brasoveanu &

Dotlačil, 2020 and Vasishth & Engelmann to apear). It is assumed that what is retrieved from declarative memory is a small, encapsulated piece of information. These pieces are called chunks and should be thought of as attribute-value matrices, or, in the parlance of ACT-R, slot-value matrices. Four examples of such chunks, corresponding to the relevant nouns from (2) and (3), are given in (4) and (5). It is assumed that the nouns have four slots: Form, (Syntactic) Function, Number, and Semantic information. These particular slots are assumed for the sake of illustration, with no claim that such a slot-value matrix exhaustively and fully captures the characteristics of these elements in memory.

$$\text{Target}: \begin{bmatrix} \text{Form} & \textit{resident} \\ \text{Function} & \textsc{Subject} \\ \text{Number} & \textsc{Singular} \\ \text{Semantics} & \textsc{Animate} \end{bmatrix} \qquad \text{Distractor}: \begin{bmatrix} \text{Form} & \textit{neighbor} \\ \text{Function} & \textsc{Object} \\ \text{Number} & \textsc{Singular} \\ \text{Semantics} & \textsc{Animate} \end{bmatrix} \tag{4}$$

$$\text{Target}: \begin{bmatrix} \text{Form} & \textit{key} \\ \text{Function} & \textsc{Subject} \\ \text{Number} & \textsc{Singular} \\ \text{Semantics} & \textsc{Inanimate} \end{bmatrix} \qquad \text{Distractor}: \begin{bmatrix} \text{Form} & \textit{cells} \\ \text{Function} & \textsc{Object} \\ \text{Number} & \textsc{Plural} \\ \text{Semantics} & \textsc{Inanimate} \end{bmatrix} \tag{5}$$

When the processor parses word after word, it builds up a syntactic parse and stores each parsed element in its declarative memory. When it encounters the verb, a subject noun parsed previously has to be retrieved from the declarative memory. The retrieval is activation-driven: all chunks are evaluated in parallel and the chunk with the highest activation is recalled. The activation of a chunk i is evaluated according to the equation in (6), where B_i is the base activation of the chunk i, S_i is the spreading activation of the chunk i, and ϵ is noise.

$$\text{ACT-R activation of a chunk in declarative memory: } A_i = B_i + S_i + \varepsilon \tag{6}$$

We will now consider B_i and S_i in detail, with an eye on how the activation captures the cue-based properties of retrieval, summarized in Section 2.1.

The base activation B_i of a chunk is given in (7). It is the log of the sum of t_k^{-d}, where t_k is the time elapsed between the time of presentation k and the time of retrieval. d is a negative exponent (decay). This is a free parameter of ACT-R, which, however, is almost always set at its default value of 0.5. "Presentation" in ACT-R means two things: (i) the chunk was created for the first time, or (ii) the chunk was recalled from memory. For example, if we are to measure the activation of the target *key*, one t_k would be the time elapsed between the creation of the noun *key*, that is, the moment the word was parsed and the structure chunk was built, and the time at which the model attempts to retrieve the noun from memory. Other t_k time elements would represent the time elapsed between previous recalls of that noun and the current recall. In this case, it is likely that there are no such other t_k times.

$$\text{ACT-R base activation: } B_i = \log\left(\sum_{k=1}^{n} t_k^{-d}\right) \qquad (d \text{ - decay, free parameter}) \tag{7}$$

The base activation decreases with the time elapsed since the presentation of the chunk. As such, it captures the decay of activation as the time progresses from the use of the chunk. It does not play a role in the interference pattern discussed in Section 2.1, and it only models how decay affects recall. For more details, see Anderson (1990) and Anderson and Schooler (1991).

The second element in the calculation of activation, spreading activation, is more relevant for us. Generally speaking, it captures the effect of the current cognitive state on retrieval. In particular, it represents the spread of activation from the current cognitive state to chunks in declarative memory. The spreading activation for a chunk i is defined in (8). It is the sum of the multiplication $W_j S_{ji}$ for every cue j that accompanies recall.

$$\text{ACT-R spreading activation: } S_i = \sum_j W_j S_{ji} \qquad (W \text{ - weight, free parameter}) \tag{8}$$

In (8), W_j is the weight for the cue j. The weight is a free parameter, with default value assumed to be proportional across cues, for example, $\frac{1}{n}$ where n is the number of cues. S_{ji} is the associative strength between the cue j and the chunk i, and formally, it is modeled as the pointwise mutual information:

$$S_{ji} = \mathbf{pmi}(i, j) = \log \frac{p(i, j)}{p(i)p(j)} \tag{9}$$

ACT-R estimates the value in (9) as follows. First, in case j is not predictive of the chunk i, it assumes that $S_{ji} = 0$. This happens, simplifying slightly, when the cue j is not present in chunk i. When the cue is present in chunk i, S_{ji} is calculated as:

$$\text{Estimated associative strength: } S_{ji} = S - \log(\text{fan}_j) \qquad (S \text{ - free parameter}) \tag{10}$$

S is the log of the size of the declarative memory, but commonly, it is hand-selected as a large enough value to ensure that S_{ji} is always positive (see Bothell, 2017). fan_j is simply the number of chunks that have the cue j as its value. The formula S_{ji} should make an intuitive sense: the associative strength will be large when j appears only in few chunks since in that case, j is highly predictive for each of those chunks; the associative strength will decrease with the increase of chunks that carry j as its value.

Finally, the formula in (11) shows how A_i, the activation of a chunk i, is related to the time it takes to retrieve the chunk i from declarative memory, T_i. The relation between A_i and T_i is modulated by two free parameters, F, latency factor, and f, latency exponent. When both parameters are set at 1 (their default value), the retrieval time of a chunk i is just the exponential of its negative activation.

$$\text{Retrieval time: } T_i = Fe^{-fA_i} \qquad (F, f \text{ - free parameters}) \tag{11}$$

Now, with this background, let us see how we capture the data summarized in Section 2.1. Before doing so, let us stress that none of the properties were constructed to describe the findings in Section 2.1. They just fall out from independently motivated properties of declarative memory and retrieval in ACT-R.

Let us start with the inhibitory interference from Fig. 1. In this case, we assume that the two cues [+ANIMATE] and [+SUBJECT] form (a part of) the current cognitive state.1 Thus, they affect retrieval through spreading activation. The spreading activations for the target and the distractor in Fig. 1 are:

$$\begin{aligned} S_{resident} &= W \cdot S_{[+\text{ANIMATE}],resident} + W \cdot S_{[+\text{SUBJECT}],resident} \\ S_{neighbor} &= W \cdot S_{[+\text{ANIMATE}],neighbor} + W \cdot S_{[+\text{SUBJECT}],neighbor} \\ S_{warehouse} &= W \cdot S_{[+\text{ANIMATE}],warehouse} + W \cdot S_{[+\text{SUBJECT}],warehouse} \end{aligned} \tag{12}$$

The spreading activation for the target, the subject *resident*, is higher than the activation for the distractors since both addends in the first equation are greater than zero. However, how high $S_{resident}$ is depends on whether we are in case (2-a) or (2-b). In (2-a), see the top figure in Fig. 1, [+ANIMATE] is shared by the target and the distractor. Since two chunks in the declarative memory carry this value, the associative strength of $S_{[+\text{ANIMATE}],resident}$ will be (assuming for the sake of concreteness that no other chunks carry the value):

$$S_{[+\text{ANIMATE}],resident} = S - log(2) \tag{13}$$

In (2-b), see the bottom figure in Fig. 1, [+ANIMATE] exclusively singles out the target. Since the value is not shared across chunks in the declarative memory, the associative strength will be higher:

$$S_{[+\text{ANIMATE}],resident} = S - log(1) = S \tag{14}$$

The activation of the chunk *resident* will be higher in (2-b) compared to (2-a) and the increased activation will result in a decrease in retrieval time, see 11. Thus, the model of declarative memory in ACT-R can capture the inhibitory interference of partially matching distractors as a case of decreased activation strength between a cue and a chunk. The decrease, in turn, is caused by the fact that the cue is shared across different chunks, that is, the fan of the cue is larger.

Let us see how the facilitatory interference from Fig. 2 is derived. In this case, the two cues forming (a part of) the current cognitive state are [+PLURAL] and [+SUBJECT] and the spreading activations for the target chunk and the distractor chunk are:2

$$\begin{aligned} S_{key} &= W \cdot S_{[+\text{PLURAL}],key} + W \cdot S_{[+\text{SUBJECT}],key} \\ S_{cell(s)} &= W \cdot S_{[+\text{PLURAL}],cell(s)} + W \cdot S_{[+\text{SUBJECT}],cell(s)} \end{aligned} \tag{15}$$

Note that $W \cdot S_{[+\text{PLURAL}],key}$ is 0 (because *key* is singular). Similarly, $W \cdot S_{[+\text{SUBJECT}],cell(s)}$ is 0 (because *cell(s)* is an object) so we can simplify (15) into:

$$\begin{aligned} S_{key} &= W \cdot S_{[+\text{SUBJECT}],key} \\ S_{cell(s)} &= W \cdot S_{[+\text{PLURAL}],cell(s)} \end{aligned} \tag{16}$$

If the distractor appears as singular, see the top figure in Fig. 2, then the associative strength of the distractor is 0 (because *cell* receives no activation from [+PLURAL]). If the distractor, however, appears as plural, see the bottom figure in Fig. 2, then the associative strength of

the distractor is greater than 0. Thus, in the bottom figure, the activation of the distractor is greater than in the top figure. This would result in decreased retrieval times if the distractor is recalled, which could happen if the activation of the distractor is higher than the activation of the target. The activation of the distractor can be higher than the activation of the target under several circumstances:

- ϵ, the noise parameter, happens to increase the activation of the distractor over the activation of the target.
- The base activation of the distractor is higher than the activation of the target, enough so that the distractor is recalled. This could happen if the distractor is more recently or very often presented/used.
- $S_{cells} > S_{key}$, enough so that the distractor is recalled. This could happen if the cue selecting distractor is very selectively tied to just that chunk or the weight for S_{cells} is higher.

When the distractor is recalled over the target, this results in faster reading times (since the distractor will have a higher activation than the target and higher activations correspond to faster recall times, see 11). Thus, any of these circumstances are enough to capture the facilitatory interference of partially matching distractors.

We see that ACT-R assumes a cue-based retrieval system that predicts a particular pattern of interferences due to distractors and the pattern is, at least to some extent, observed in the resolution of dependencies.

We will now turn to the parsing system that can leverage this organization of memory. In Section 3, it is shown that there is a class of parsers (transition-based parsing) that can be directly built as a case cue-based retrieval model.

3. Transition-based parsing

In this section, transition-based parsers are introduced. As we will see, the parsers are compatible with the memory structures discussed in Section 2 and can be, to a large extent, embedded in ACT-R. This embedding will be tested in the following sections.

Transition-based parsers are parsing systems that predict transitions from one state to another, following decisions made by a classifier. Since the classifier plays a crucial role in this type of parsers, these parsers are also sometimes called classifier-based parsers.

Transition-based parsers are most commonly implemented for dependency grammars, and arguably, they are most successful and widespread when constructing dependency graphs (Nivre et al., 2007) but they have also been applied to phrase structure parsing (Kalt, 2004; Sagae & Lavie, 2005), including neural phrase-structure parsing (Kitaev & Klein, 2018; Liu & Zhang, 2017). This paper also implements transition-based parsing for a phrase-structure parser. We will look at a shift reduce variant of the transition-based parsing algorithm, which is arguably the most common type of transition-based parser for phrase structures and also comes closest to the transition-based parsing of dependency graphs (see Sagae & Lavie, 2005).

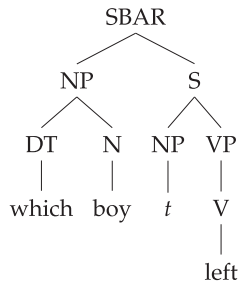


Fig. 3. Phrase structure of *which boy left?*.

3.1. Algorithm of transition-based phrase-structure parsing

The parsing algorithm works with two databases, a stack of constructed trees $\mathcal{S}$ and a stack of upcoming words with their part-of-speech (POS) tags $\mathcal{W}$. When parsing begins, $\mathcal{S}$ is empty and $\mathcal{W}$ carries the upcoming words as they appear in the sentence, so that the first word appears at the beginning of the stack, followed by the second word, etc.

Parsing proceeds by selecting actions based on the content of $\mathcal{S}$ and $\mathcal{W}$. Every parsing step $\mathcal{P}$ is a function from $\mathcal{S}, \mathcal{W}$ to actions $\mathcal{A}$, $\mathcal{P} : \mathcal{S} \times \mathcal{W} \rightsquigarrow \mathcal{A}$. Broadly speaking, three actions could be taken by the parser:

- shift
- reduce
- postulate gap

The first action, *shift*, pops the top element from the stack $\mathcal{W}$ and pushes it as a trivial tree onto stack $\mathcal{S}$. The element in $\mathcal{W}$ is a pair $\langle$word, POS$\rangle$, the tree moved onto the stack is just the POS tag with the terminal being the actual word.

The second action, *reduce*, pops the top element (if the reduction is unary) or it pops top two elements (if the reduction is binary) in the stack of constructed trees $\mathcal{S}$ and creates a new tree. If the reduction is unary, the new tree has just one daughter under the root, the tree that was just popped from the stack. If the reduction is binary, the newly created tree has two daughters, the two trees that were just popped from the stack. In either case, the newly constructed tree is pushed on top of the stack $\mathcal{S}$ and it is specified what label the root of the tree has. It is assumed that all trees are at most binary, so no further reductions beyond binary reductions are necessary.

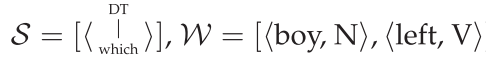
Finally, the third action, *postulate gap*, postulates a gap and resolves it to its antecedent.3

There are several restrictions on the three actions. First, no shift can be applied when $\mathcal{W}$ is empty. When $\mathcal{S}$ is empty, no reduce can be applied and when it has only one tree, reduce binary cannot be applied. Finally, no more than two postulate gaps actions can be applied between two shifts. This last restriction ensures that the system does not fall into the infinite regress of postulation gaps.

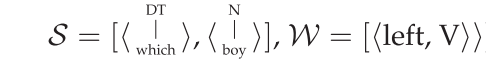
Let us consider a simple example: parsing of *which boy left?*. The phrase structure is shown in Fig. 3 and the parsing steps are:

1. Starting position: $\mathcal{S} = [], \mathcal{W} = [\langle \text{which, DT} \rangle, \langle \text{boy, N} \rangle, \langle \text{left, V} \rangle]$

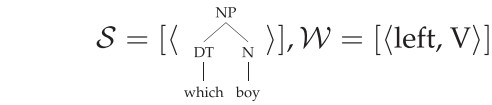
2. shift



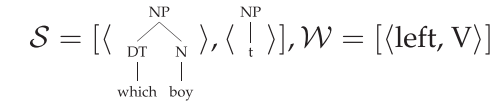
3. shift



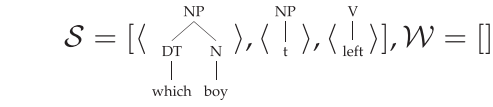
4. reduce (binary) with label NP



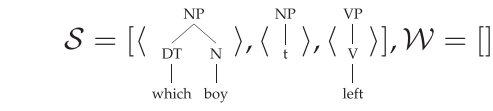
5. postulate gap



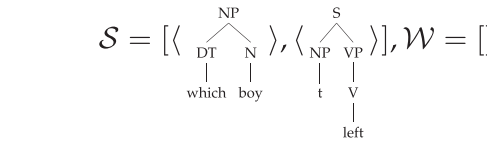
6. shift



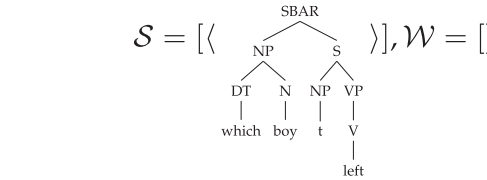
7. reduce (unary) with label VP



8. reduce (binary) with label S



9. reduce (binary) with label SBAR



In this illustrative example, we assume that the parser knows what the right phrase structure is and parses toward that structure. Of course, the interesting question is what happens when the phrase structure is unknown and the parser needs to decide what action to take. This is where cue-based retrieval becomes relevant.

3.2. Parsing steps as memory retrievals

Generally speaking, the parsing step has to decide which action (among *shift*, *reduce* and *postulate gap*) should be taken, and, if *reduce* is selected, how should the reduction be done: should it be unary or binary? What should the root label of the newly constructed tree be?

This is the point at which transition-based parsing developed in computational linguistics meets memory systems established in psycholinguistics. We will assume and test the

following linking hypothesis:

> Linking hypothesis between parsing and memory:
>
> A parsing step is a cue-based retrieval from declarative memory. The retrieval uses as cues the information from $\mathcal{S}$ and $\mathcal{W}$ and the retrieved chunk specifies the action (from actions in $\mathcal{A}$) that should be taken as the parsing step. (17)

Why should the linking hypothesis hold? Because of the way, learning works in ACT-R. When language users are at some parsing step X, they are aware of the current context, represented by $\mathcal{S}$ and $\mathcal{W}$. Their goal is to select the right action at that moment. Let us say they select one such action, fulfilling the goal of deciding what parsing step to take. This parsing step, consisting of the context and the action, is then stored as a chunk in declarative memory and can be recalled in the future to guide the same user through parsing steps with similar context. This is arguably the most common line of how ACT-R agents learn (Anderson & Lebiere, 1998; Lebiere, 1999).

While it might be possible to think of the context as complete trees in $\mathcal{S}$ and all information in $\mathcal{W}$, we will limit the amount of information in the two databases. It will be assumed that $\mathcal{S}$ and $\mathcal{W}$ carry only some features about the trees/upcoming words. The features are listed in (18). Thus, the parser itself never has a full snapshot of the phrase structure that it is deriving. It only carries some minimal, local information. The phrase structure can always be reconstructed through parsing steps the ACT-R agent (and humans) took but there is no single snapshot in which all the information is available to the agent. This position is common in ACT-R parsing, see, for example, Lewis and Vasishth (2005).

Features representing context: (18)

a. 0, 1, or 2 upcoming words with their POS (see more on this below),
b. root labels of top four elements in $\mathcal{S}$,
c. lexical head and the POS of the lexical head for top four elements in $\mathcal{S}$,
d. left and right children in top two elements in $\mathcal{S}$, and
e. antecedent carried (yes or no).

These features should be easy to understand, maybe with the exception of the antecedent carried and lexical head. The antecedent-carried feature has only two possible values, yes or no. It is set to yes when an element has been parsed that needs to be resolved through a gap postulation and the gap has not yet been postulated. In this paper, it is assumed that only wh-phrases need to be resolved. That means that wh-phrases will be the only element that will form dependencies in the upcoming case studies. The lexical head is a terminal that projects its phrase (a verb is the head of a verb phrase, a noun is the head of an NP, etc.) and is relevant even beyond the phrase (e.g., verbs are heads of clauses, *S*; see Collins, 1997 on head

projection in computational parsers, which this works follows). We will store lemmatized lexical heads.

As an example, assume that the sentence *which boy left?* is the only sentence parsed and stored in declarative memory. Then the parsing memory would solely include the parsing steps listed above. For instance, the parsing step *reduce (binary) with label SBAR* would be stored in declarative memory as shown in (19). Only the slots that carry a value are listed.

Last parsing step of *whichboyleft*? stored in declarative memory:

$$\begin{bmatrix} \text{root label of top element in } \mathcal{S} & S \\ \text{root label of 2nd element in } \mathcal{S} & NP \\ \text{lex. head of top element in } \mathcal{S} & leave \\ \text{POS of head of top element in } \mathcal{S} & V \\ \text{lex. head of 2nd element in } \mathcal{S} & boy \\ \text{POS of head of 2nd element in } \mathcal{S} & N \\ \text{left child of top element in } \mathcal{S} & NP \\ \text{right child of top element in } \mathcal{S} & VP \\ \text{left child of 2nd element in } \mathcal{S} & DT \\ \text{right child of 2nd element in } \mathcal{S} & N \\ \text{antecedent carried} & no \\ \text{action} & \textit{reduce (binary)} \\ \text{label} & SBAR \end{bmatrix} \tag{19}$$

As one can see, the parsing step chunk stores the action (e.g., *reduce*) and the corresponding label (*SBAR*) along with the context in which the action took place. When parsing a novel context, the retrieval will be attempted. The context will spread activation to parsing steps chunks in the declarative memory and select such a chunk that has the highest activation. For instance, assume that the current context is in (20). This context could represent, for example, almost finished parsing of the sentence *which woman dances?*.

Example context

$$\begin{bmatrix} \text{root label of top element in } \mathcal{S} & S \\ \text{root label of 2nd element in } \mathcal{S} & NP \\ \text{lex. head of top element in } \mathcal{S} & dance \\ \text{POS of head of top element in } \mathcal{S} & V \\ \text{lex. head of 2nd element in } \mathcal{S} & woman \\ \text{POS of head of 2nd element in } \mathcal{S} & N \\ \text{left child of top element in } \mathcal{S} & NP \\ \text{right child of top element in } \mathcal{S} & VP \\ \text{left child of 2nd element in } \mathcal{S} & DT \\ \text{right child of 2nd element in } \mathcal{S} & N \\ \text{antecedent carried} & no \end{bmatrix} \tag{20}$$

The model would then very likely retrieve (19), since the current context will spread the activation from almost all features but lexical heads to (19), and no other parsing step chunk from the sentence *which boy left?* will receive a comparable boost in spreading activation.

Under this view, undertaking a parsing step is a case of memory retrieval that follows the rules in Section 2.2. Consequently, it is predicted that parsing will be activation-driven and different parsing steps might require different amounts of time depending on the time it takes to retrieve them. Parsing steps with higher activations will be recalled faster than parsing steps with lower activations. Activations, in turn, are affected in the exactly same way as any other case of cue-based retrieval in ACT-R.

3.3. Computational model of transition-based parser, training, and accuracy

To test the cue-based model of parsing, we consider a concrete declarative memory structure with chunks that represent correct past parsing steps. For our purpose, we use Penn Treebank (Marcus, Marcinkiewicz, & Santorini, 1993). As is standard, we split the section of Penn Treebank as follows: all the sections up to and including section 21 are used to train the parser, that is, to collect the correct parsing steps; section 22 is used for development; section 23 is used to test the accuracy of the parser. Before training we pre-process and prepare the phrase structure by (i) transforming phrases into binary structures in the way described in Roark (2001) (see Roark, 2001; Sagae & Lavie, 2005 on why this is needed), (ii) annotating phrases with head information, (iii) removing irrelevant information (coreference indices on phrases),4 and (iv) lemmatizing tokens so that lexical heads are stored as lemmas, not as inflected tokens.

Parsing of novel sentences consists of recalling the chunks from the declarative memory in the order of their activation. To calculate activation for each chunk, formulas in Section 2.2 are applied. The parser recalls three chunks with the highest activations and the action that has the highest activation, summed up over the three recalled chunks, is carried out.5

Even though it is not the goal of this paper to study the accuracy of the parser, it might be of interest that when tested on section 23, the parser shows Label Precision as 70.2, Label Recall as 72.4, and F1 as 71.3. When we restrict attention to sentences of 40 words or less, as is common, Label Precision is 73.7, Label Recall is 75.9, and F1 is 74.8.6 While these precision and recall values are far away from the current state of the art,7 this level of accuracy is sufficient for the modeling of the experimental items to which we now turn.

4. Modeling reading data

We will now go through the evidence for the cue-based model of parsing. Three cases will be discussed.8

Case 1 and case 2 are reading experiments and case 3 consists of modeling self-paced reading corpus data. In case 1 and case 2, we will see that the parser can be combined with a few extra assumptions for reading to generate reading latencies that fit the actual data. In case 3, we will see that the activations of parsing steps are good predictors for reading measures.

4.1. Case 1: Retrieval of dependents and retrieval of processing steps

We start by modeling reading data from Experiment 1 in Grodner and Gibson (2005) (also used in Lewis & Vasishth, 2005). This is a self-paced reading experiment (non-cumulative moving-window; Just, Carpenter, & Woolley, 1982). Participants read word-by-word sentences in which the subject NP is modified by a subject or object extracted relative clause (RC). A subject-gap example is provided in (21-a), and an object-gap in (21-b). *t* signals a gap and it appears in the position in which it would be postulated according to Penn Treebank annotation rules and standard assumptions in linguistics. The gap shows where the dislocated argument, the relative pronoun *who*, is interpreted.

a. The reporter who *t* sent the photographer to the editor hoped for a story.

b. The reporter who the photographer sent *t* to the editor hoped for a story. (21)

There are six regions of interest (ROIs) that we model, underlined in the examples above. The ROIs start at the first word of the relative clause and stop at the penultimate word of the relative clause.9

Grodner and Gibson (2005) have been chosen for several reasons. Parts of their data have been simulated by the first explicit linguistic model of ACT-R, Lewis and Vasishth (2005), and played a role in other cognitive models of reading (e.g., Chen & Hale, 2021). It is good to see that our model can replicate their results. Not only that, we will see that our model can also significantly extend the findings of Lewis and Vasishth (2005). Lewis and Vasishth (2005) studied only the difference between reading times on the verbs in (21-a) and (21-b), while our model will be able to simulate actual reading times, not just differences between conditions, and it will do so for 12 words in total. Moreover, (21) is an interesting case in which parsing interacts with another aspect of cue-based retrieval, the recall of dependents (wh-words). By simulating Grodner and Gibson (2005) we will have evidence that different forms of retrieval, be this wh-dependency in relative clauses of parsing steps, can be modeled by one and the same mechanism: cue-based retrieval.

In Sections 4.1.1 and 4.1.2, it is shown how the parser can be combined with other components of reading, and in Section 4.1.3, we inspect what syntactic predictions the parser makes. In Section 4.1.4, it is shown how the model can be fit to reading times through the estimation of ACT-R free parameters. In Section 4.1.5, we turn our attention to two other models that ignore or modify the parsing component of the model and see that the changes result in a worse fit. That is, we will see that not only can our model fit the data, but slight modifications in the parsing component degrades the fit, suggesting that the parsing component as proposed is needed for the modeling of reaction times.

4.1.1. Sequential model for reading

The cue-based model of parsing has been specified in Sections 2 and 3. The procedure goes as follows. When the parser is at word *n* and a parsing step needs to be carried out, the parser retrieves three best fitting chunks from the declarative memory ordered by activation (calculated as a sum of base activation and spreading activation) and applies the most common

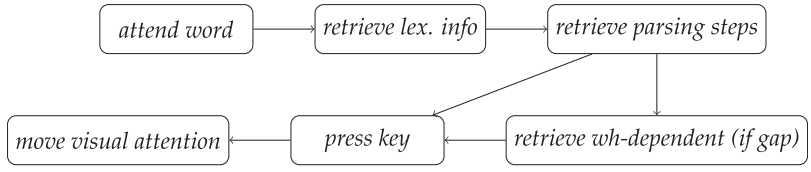


Fig. 4. Sequential model of reading on one word. Each box represents one subprocess. The arrows represent the order of subprocesses. There are two arrows from *retrieve parsing steps* because *retrieve wh-dependent* is not always triggered (only when a gap is postulated by the parser).

action shared by the three chunks. In case of a tie, the action from the chunk with the highest activation is used. The parser repeats this procedure until it encounters shift. At that moment, the parser is done with integrating word n and can move its attention to word $n + 1$.

In self-paced reading that we are about to model, readers, however, do much more than just retrieving and applying parsing steps. It seems uncontroversial that a model simulating reading should, at very least, attend to visual information on word n, retrieve lexical information on that word, parse, press a key (to reveal the next word), and move visual attention to the next word. To have a chance at having a descriptively correct computational model, we should add at least these components.

Each of the listed steps is a different process with its own properties. The processes are linked together and controlled by the procedural knowledge in ACT-R. We see how the processes fire one by one on a word n in Fig. 4. It is assumed that these processes are repeated on every word. Postulating these sequential steps for self-paced reading is relatively standard (see Brasoveanu & Dotlačil, 2020; Lewis & Vasishth, 2005). Firing each of the processes takes the same amount of time in the procedural system, specified in (22):

$$\text{Rule firing in ACT-R: } r(r \text{ is a free parameter, default—50 ms}). \tag{22}$$

In addition to that, submodules involved in each process can incur extra processing time.

The process *attend word* visually attends to a word. To keep the model simple, I will not try to model any details of visual attention, just assume that visual attention takes the fixed amount of time, in line with basic/default models of ACT-R (Bothell, 2017). It is assumed that attending takes 50 ms, the default value of rule firing in ACT-R. The processes *retrieve lex. info*, *retrieve parsing steps*, and *retrieve wh-dependent* will be discussed below. This leaves us with *press key* and *move visual attention*. *Press key* is modeled assuming the basic model of motor actions in ACT-R, which is inspired by the EPIC cognitive architecture (Bothell, 2017). It is assumed that readers have their finger prepared on the key to be pressed. In that case, the simple model of motor actions in ACT-R, followed here, postulates that it takes 150 ms to press the key. Crucially, during this time, the procedural system is free to carry out any other actions in the sequential model. That means that moving visual attention can happen concurrently with key presses. Since attending the next/upcoming word in the sentence should not take more than 150 ms, I will assume that moving visual attention does not add any extra time beyond 150 ms required by the motor module.

4.1.2. Retrieval from declarative memory

Let us now go back to the three processes that involve declarative memory and retrieval therefrom: *retrieve lex. info*, *retrieve parsing steps*, and *retrieve wh-dependent*. These processes take r amount of time each, but aside from that we want to know how much time it takes to retrieve an element. All relevant equations to calculate the retrieval time have been given in Section 2.2. Let us repeat that the retrieval time is a function of activation of a retrieved chunk and modulated by two free parameters, (23-a). Activation is calculated as the sum of base activation and spreading activation, (23-b). (We ignore the noise parameter ϵ, so that retrieval time becomes deterministic.) Base activation and spreading activation are repeated in (23-c) and (23-d).

$$\begin{array}{lll} \text{a.} & T_i = Fe^{-fA_i} & (F,\ f \text{ - free parameters}) \\ \text{b.} & A_i = B_i + S_i & \\ \text{c.} & B_i = \log\left(\sum_{k=1}^{n} t_k^{-d}\right) & (d \text{ - parameter set at } 0.5) \\ \text{d.} & S_i = \sum_j W_j \cdot (S - \log(\text{fan}_j)) & (W,\ S \text{ - free parameter}) \end{array} \tag{23}$$

Note that, when a chunk does not share any cues with the context, S_i becomes zero and can be ignored. The recall of syntactic information is driven by context cues and so is the recall of wh-dependents but the lexical retrieval has no cues that are of interest and for this reason, it is assumed that spreading activation is zero for this case of retrieval. It is most likely a simplifying assumption, but as we will see, it does not harm the fit of the model.

The cues used for the spreading activation of parsing were described in 18. Since we now deal with self-paced reading, in which readers have no look-ahead possibility, it is assumed that no upcoming words are used as context cues (see 18-a). For the wh-recall, only the syntactic category of the wh-dependent is used as a cue to increase spreading activation (more could be added; see Arnett & Wagers, 2017; Kush, 2013; Kush et al., 2015; Patil, Vasishth, & Lewis, 2016; Parker & Phillips, 2017; Smith & Vasishth, 2020 for investigations of what features are relevant for cue-based retrieval).

The parameter d from (23-c) is set at its default value, 0.5 (see Anderson & Lebiere, 1998), and S from (23-d) is set at 20, which is high enough to ensure that $S - \log(\text{fan}_j)$ is always positive for any j appearing in data (see Section 2.2). Apart from d and S, the formulas in 22 and (23) have four parameters: F, f, r, W_j. These will be estimated according to the procedure described in Section 4.1.4.

Before we turn to that, we need to decide another thing: how is t_k from (23-a) found? For a retrieval of wh-dependent, this is easy: it is the time elapsed between parsing a wh-dependent, that is, parsing *who* in 21, and postulating a gap, that is, at the subject position in (21-a) or at the object position in (21-b).

For lexical retrieval and parsing steps retrieval, we estimate t_k in (23-a) from the frequency of words/parsing steps. The frequency of words is estimated from the British National Corpus. The frequency of parsing steps is estimated using the Penn Treebank corpus. The frequencies can be transformed into t_k according to the procedure described in Reitter, Keller, and

Moore (2011), Dotlačil (2018), and Brasoveanu and Dotlačil (2020). The procedure is summarized in Appendix A.

Finally, we need to clarify one last issue. At each word, parsing is finished when *shift* is recalled, at which point the processes following parsing take place, see Fig. 4. However, the retrieval of parsing steps can consist of several parsing steps, and in this way, parsing differs from the retrieval of lexical information and the retrieval of wh-dependents, which usually retrieve only a single element per word.

We could assume that retrieving each parsing step is a process in the sequential model on its own: that is, there could be several *retrieve parsing steps* processes per word. This position would be in accordance with ACT-R, which assumes the serial order in the procedural system if the same process type is involved.

However, there is a serious drawback to letting every parsing step be a process on its own. If each parsing step would correspond to one process, we would predict that reading times linearly increase with the number of parsing steps (see discussion in Section 4.1.1). We do not want to go this route, for three reasons. First, we would add another factor that would affect reading times based on syntactic properties and this effect might completely obscure our main point of investigation, the role of memory in parsing. Obviously, it is preferable to not introduce confounds into our model. The second problem is that our results would become highly dependent on the type of parsing algorithm. We make use of the shift-reduce (bottom-up) parsing algorithm. In this algorithm, steps accumulate at the end of a phrase, so we would expect that ending phrases increases reading times. Top-down parsers accumulate parsing steps when a new phrase is started and generalized left-corner parsers can accumulate steps anywhere between these two extremes (Hale, 2014; Resnik, 1992). But it is not of our interest to investigate whether one algorithm is correct, rather, we want to see whether the transition-based parsing with the linking hypothesis 17 can be fit to data. Finally, it has been proposed that often repeated parsing steps are merged/compiled into one step through production compilation (Hale, 2014), so treating them as separate would probably be empirically inadequate (and too simplistic) even if we knew what the right parsing algorithm is. I will come back to this last issue in Section 5.2.

For the just-listed reasons, another solution will be adopted. We assume that there is just one process, the retrieval of parsing steps, and the retrieval time is calculated based on the average activations of all the parsing steps recalled on that word. Other, more sophisticated relations between the number of parsing steps and the actual retrieval time have to be left for future investigations.

4.1.3. Symbolic syntactic predictions of the model

The syntactic parser constructs the correct phrase structure for the sentence, including the correct postulation of gaps for the subject and object relative clauses, see Fig. 5. As far as I know, this is the first data-driven parser that is built using assumptions of cue-based retrieval and, to a large extent, is compatible with the ACT-R cognitive architecture, yet it is able to parse sentences of this complexity correctly without any hand-coding of the syntactic rules—the whole structure is generated by the data-driven transition-based parser.

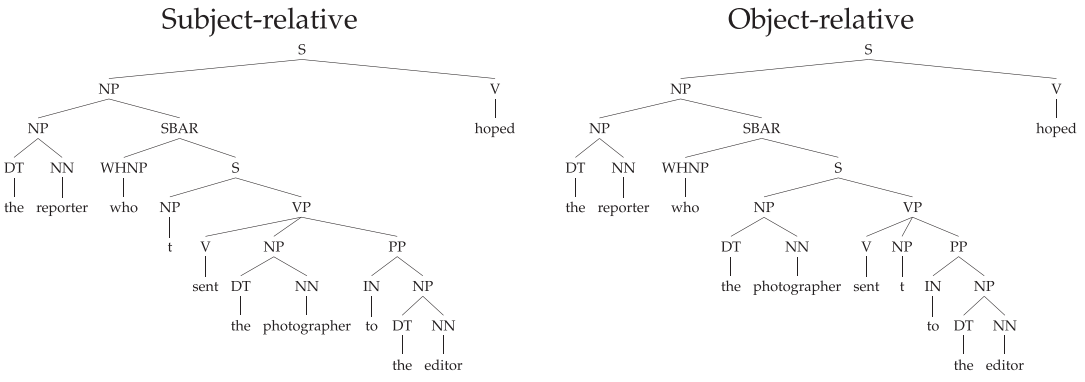


Fig. 5. The syntactic structure built by the parser. For readability, we transform binary trees into more common, n-ary versions.

It is instructive to investigate how this parsed structure comes about. The full derivation is spelled out step by step in Appendix B. Here I only focus on wh-words and gaps since these are the crux of the investigation of Grodner and Gibson (2005) and Lewis and Vasishth (2005). The following is observed. When the wh-word *who* has been just parsed, the parser, which lacks any look-ahead possibility, assumes that it just entered a relative clause and postulates a subject gap. This is due to the fact that the parser relies on past parsing steps (collected from the PTB) and subject-relative clauses are most common types in the corpus (and arguably, English). When the relative clause turns out to be the subject-relative, the gap is postulated correctly and the transition-based parser does not attempt to postulate any gaps further downstream. However, when the relative clause is not the subject-relative, the parser again tries to discharge the dependency and guesses after processing the verb that the object gap should be postulated. This postulation of the gap is immediately followed by the retrieval of the wh-element. The predictions are summarized in Fig. 6. The figure shows that the parser predicts gaps, in accordance with the theory of the Active Filler Strategy (Crain & Steedman, 1985; Frazier, 1987). The parser also matches the modeling assumptions of Lewis and Vasishth (2005), which derive slowdown in reading times of object-relative verbs by letting their ACT-R parser retrieve a wh-dependent at that position. However, unlike Lewis and Vasishth (2005) and its extension, Engelmann et al. (2013) and Engelmann (2016), this behavior of the parser is not manually created. The Active Filler Strategy is not assumed, it falls out as a consequence of the data-driven parsing and the fact that the cue-based retrieval at these positions favors *postulate gap*.

Since the parser explores only one path, it would incorrectly predict that the wrong postulation of the gap in the object-relative clause at the subject position cannot be recovered from. To avoid this, a minor correcting behavior of the parser will be assumed. It is assumed that the parser checks at each word whether the structure postulated at the previous word is compatible with the new evidence (the new word). If not, the parser will reanalyze toward the new structure and continue in parsing. This means that the parser will reanalyze at the next word after *who* in object relative clauses and will remove the gap.

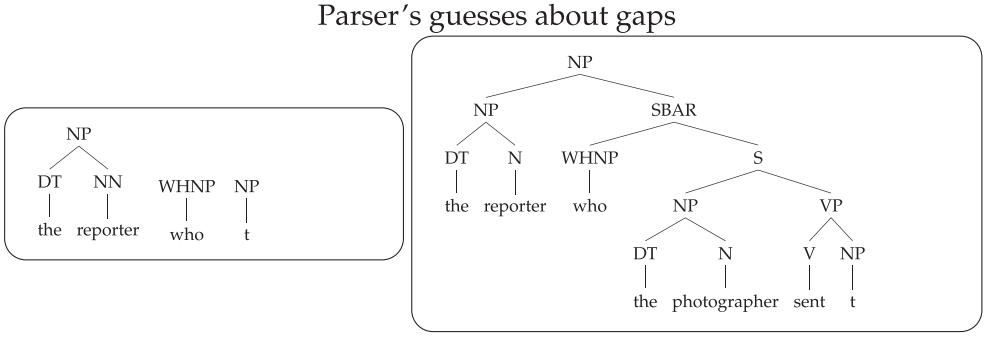


Fig. 6. Two selected steps during parsing. After parsing the wh-word, the parser guesses that a gap should be postulated for the subject position (left side). This is correct for subject relative clauses, incorrect for object relatives. For the latter, when the parser moves to the next word (*the*), it reanalyzes to the correct structure with no gap. After parsing the verb, the parser postulates a gap at the object position (right side). More details about the parser's incremental construction can be found in Appendix B.

The reanalysis itself is simplified. The parser simply takes over the phrase structure that it postulated based on the new evidence. It is assumed that the reanalysis incurs extra cost, as large as the time any subprocesses takes in the procedural system (the r parameter in 22).

Another way to understand the parser's predictions is that it expects subject-relative clauses by default and switches to object-relative clauses only when the original expectation turns out wrong. Due to the cost of reanalysis, the parser thus has the ability to predict processing difficulties for object-relative clauses as a consequence of invalid expectations. Crucially, the prediction is generated by the memory system that can also predict processing costs of object-relative clauses due to the retrieval of the wh-dependent. Thus, we can derive two types of costs, an expectation-based cost and a wh-retrieval-based processing cost within one memory system, cue-based retrieval (see Staub, 2010 for arguments that both types of processing difficulties are observed in relative clauses).

4.1.4. Bayesian modeling

There are four parameters that we need to model to fit the reader to the data: F, f, r, W_j. We will estimate them using Bayesian techniques. One should think about the Bayesian model that we consider as a Bayesian data analysis model that is used to provide the best fit of our cognitive model to the data of Grodner and Gibson (2005).

I assume the structure of the model as shown in Fig. 7. In this graph, which follows notational conventions of Kruschke (2011), the top layer represents priors, the bottom part is the likelihood. The actual data that we try to model are mean reading time per region 3–8 in subject-relative and object-relative clauses.10 To calculate the likelihood, we run all stimuli from Grodner and Gibson (2005) using priors and the model described in Sections 4.1.1–4.1.3. We collect all reaction times per words 3–8 and take the mean; the mean is the Latency variable in the part described as ACT-R(F; f; r; W_j) in Fig. 7. The Latency serves as the mean of the likelihood of the model, which is a normal distribution with standard deviation being

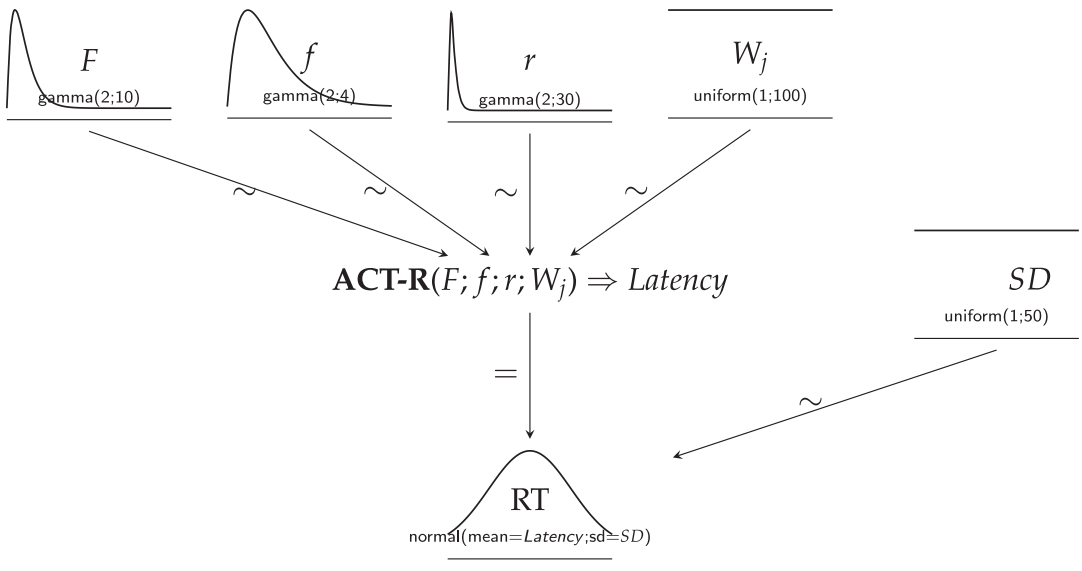


Fig. 7. Bayesian model for parameter estimation of Grodner and Gibson (2005).

estimated as another parameter, *SD*. This last parameter is not part of the ACT-R model. The likelihood can be seen in the bottom part in Fig. 7. A similar way of modeling was successfully applied in Dotlačil (2018), Brasoveanu and Dotlačil (2018), Brasoveanu and Dotlačil (2019), and Brasoveanu and Dotlačil (2020). See Dotlačil, 2018 for reasons why it is preferable to use this method rather than rely on default values of ACT-R and partially tweak them by hand selection.

The following prior structure for the ACT-R parameters is assumed:

- $F \sim Gamma(\alpha = 2, \beta = 10)$
- $f \sim Gamma(\alpha = 2, \beta = 4)$
- $r \sim Gamma(\alpha = 2, \beta = 30)$
- $W_j \sim Uniform(1, 100)$

SD, the parameter to model standard deviation of the likelihood, has the following prior:

- $SD \sim Uniform(1, 50)$

The priors for the first two ACT-R parameters have the mean values of 0.2 and 0.5, respectively, but, roughly speaking, the distributions are broad enough to not exclude any value between 0 and 1. Values in the range 0–0.3 are most likely but extremely low values are penalized. This takes into account previous findings that F and f, modulating retrieval times, are in language models almost always below 0.5 but not exceedingly small (Brasoveanu & Dotlačil, 2020). The third prior, r, has the mean of 0.05 (seconds). This is the default value for r in ACT-R. Finally, the prior for W_j, measuring the weight of associative strength between a cue and a chunk, is set as a uniform distribution that takes any values between 1 and 100 as

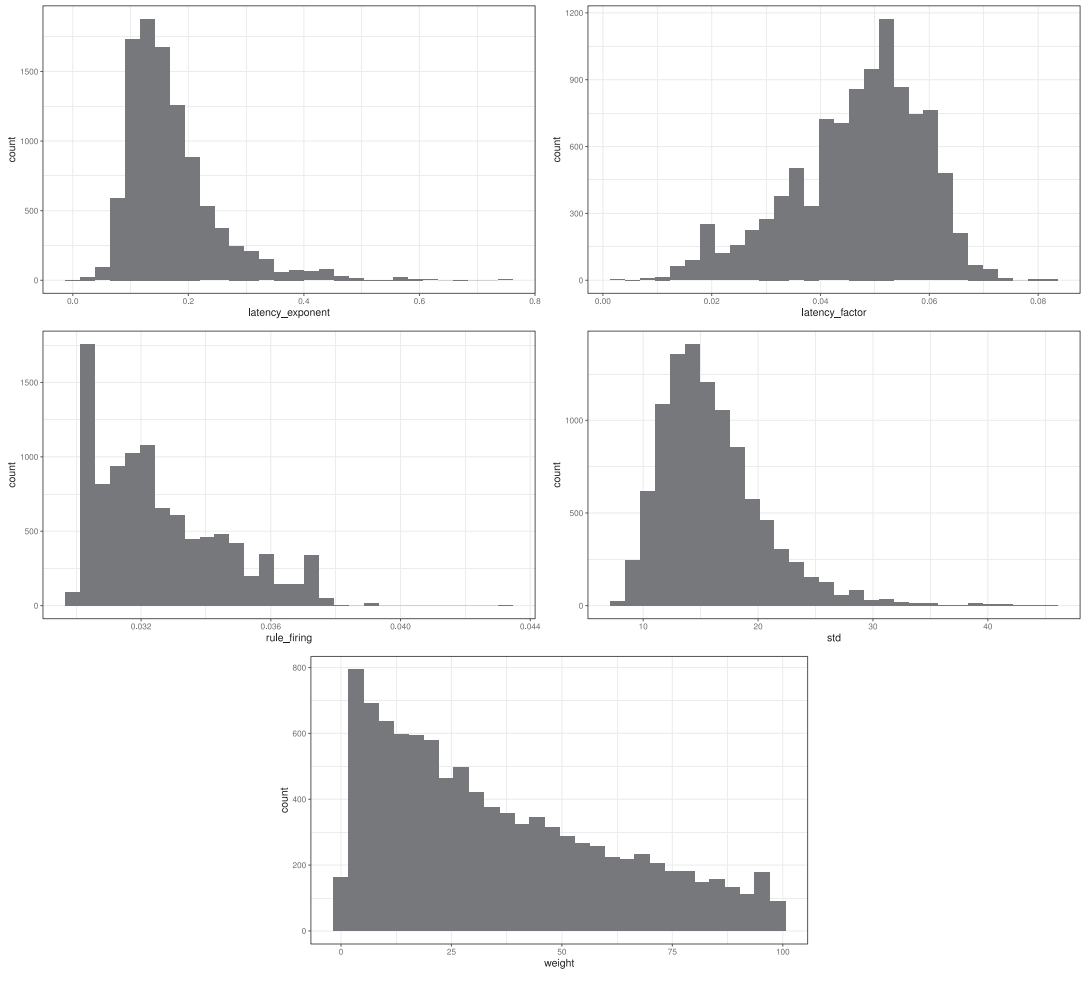


Fig. 8. Posteriors for the five parameters estimated in the Bayesian ACT-R model.

equally likely. This flat prior takes into account that we have very little evidence a priori how cues are weighed for the retrieval of parsing steps and wh-dependents.

The estimation is done using PYMC3 and MCMC-sampling with 5,500 draws, 2 chains, and 500 burn-in. The Rhat values (Gelman et al., 2013) for the four parameters are below 1.05, showing that the chains have converged. More details about the model are given in Appendix C.

4.1.5. Results

Let us first summarize the posterior distribution of the modeled parameters. See also Fig. 8:

- F—median: 0.05, sd: 0.01

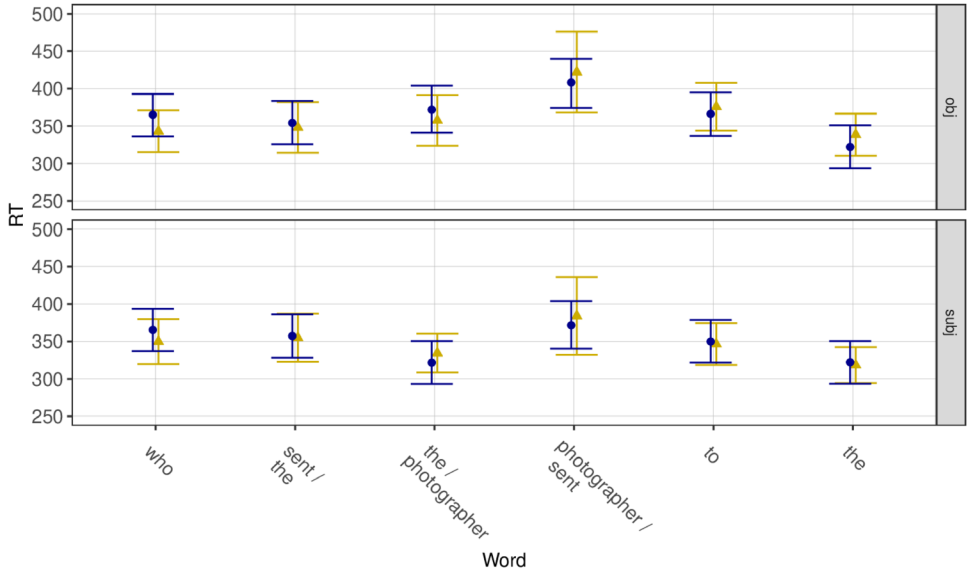


Fig. 9. Model 1 of reading—posterior predictive. The blue dots are predicted mean RTs and the blue bars provide the 95% credible intervals. The observed data are in yellow. The yellow triangles are observed mean RTs, and the yellow bars are $+/-$ 2 standard errors, taken from Grodner and Gibson (2005).

- f—median: 0.15, sd: 0.08
- r—median: 0.03, sd: 0.002
- W_j—median: 29, sd: 27
- SD—median: 15, sd: 5

The posterior values of the first three parameters are not far off from previous estimations in psycholinguistics (Brasoveanu & Dotlačil, 2020).

The posterior predictive distribution of the model is of the main interest. We want to see what our model predicts as mean reaction times and whether this fits the observed data. The predictions are plotted against the observed data in Fig. 9. The yellow triangles indicate the observed mean RTs for each word, the yellow bars indicate $+/-$ 2 standard errors (means and SEs taken from Grodner & Gibson, 2005), the blue segments provide the 95% CRIs (credible intervals) for the mean RTs predicted by the Bayesian model, and the blue dots are the predicted mean RTs. The 95% CRIs cover the observed mean RTs, and moreover, the observed mean RTs are often close to the mean RTs of the model. That is, we see that the parameters can be estimated in such a way that the model very closely fits the data.

Two things should be kept in mind in the evaluation of the model. First, all the parameters affect reading times of every word and most of the parameters affect multiple processes at the same time (e.g., F will affect lexical retrieval times, retrieval times of wh-dependents, and retrieval times of parsing steps). Yet, the parameters are estimated only once for all the processes and for the full run through the experiment—they are not estimated word by word and not process by process. Furthermore, in contrast to almost all previous works on ACT-R

and linguistics (see Section 5.2 for a detailed comparison), there is almost no space for hand-coding of the model. The only part of the model that is manually created is the sequence in Fig. 4, that is, the handful of the rules and their order. The translation of this sequence into reading times is derived by the computational cognitive architecture ACT-R, the parse is constructed by a transition-based parser (embedded in ACT-R), and the parameter estimation is generated by a Bayesian model.

4.1.6. Syntax-free models of self-paced reading

We see that the model developed in Section 4.1.1–4.1.3 can approximate mean RTs reasonably well. We now check whether the syntactic parser, which is the main point of this investigation, is at least partially responsible for this success. We investigate this question by comparing the model to two other models.

In Model 2, it will be assumed, contrary to the symbolic predictions of the transition-based parser, that no Active Filler Strategy is present. That is, the parser will still retrieve parses but it will not postulate a gap at the wh-word/verb, rather, the parser will wait for the unequivocal evidence to do so. Let us see concretely what that means on the example sentences from Grodner and Gibson (2005), repeated from above:

(24)
a. The reporter who *t* sent the photographer to the editor hoped for a story.
b. The reporter who the photographer sent *t* to the editor hoped for a story.

If the parser waited with gap postulation, it would only posit the gap in the subject-relative clause, (24-a) when it reads the verb. For (24-b), the parser assumes a gap when it reads the preposition. That is, Model 2 is manually set to retrieve wh-dependents at different positions that the transition-based parser based on its learning corpus does.

In Model 3, the syntactic component is completely switched off. This means we omit the step *retrieve parsing steps* in Fig. 4 (and with that, we also have to omit *retrieve wh-dependent*, since that step is dependent on the triggering of gap postulation).

Apart from these changes, Model 2 and Model 3 are exactly the same as the first model.

We now estimate the same parameters as for the first model and study posterior predictions for reading times per words 3–8. The posterior predictions are given in Figs. 10 and 11.

We can see that both models are good enough to capture the general trend in the data. This should not be very surprising since the models still include lexical retrieval and other basic components, so that reading times can be approximated quite well. However, both models have a worse fit than Model 1.

For Model 2, see Fig. 10, we see that the model fails at object relative clauses at the verb and the preposition. It is too fast on the former word since it does not postulate a gap, unlike Model 1, and too slow for the second word, since it postulates a gap, unlike Model 1. Compared to Model 1, this model also overestimates reading times for the subject-relative clause on the verb, which comes about because it tries to resolve the wh-dependency at this point, which will slow it down. In general, the model fails precisely in positions that we would expect it to fail. To be sure, the 95% credible intervals of posterior predictive distribution include most mean RTs, but this is at the cost of less precise posterior distributions for reading times, as can

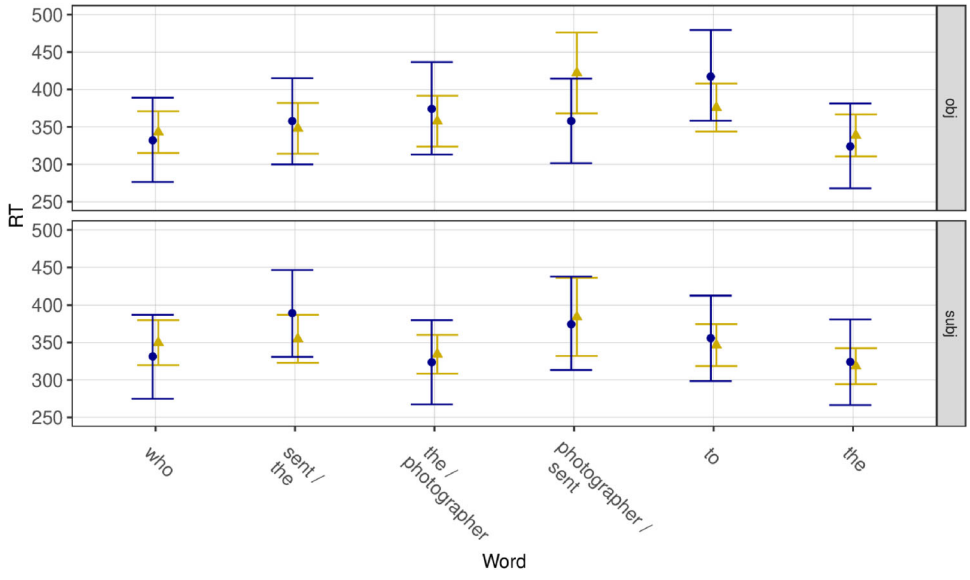


Fig. 10. Model 2 of reading—posterior predictive. The blue dots are predicted mean RTs and the blue bars provide the 95% credible intervals. The observed data are in yellow. The yellow triangles are observed mean RTs, and the yellow bars are +/− 2 standard errors, taken from Grodner and Gibson (2005).

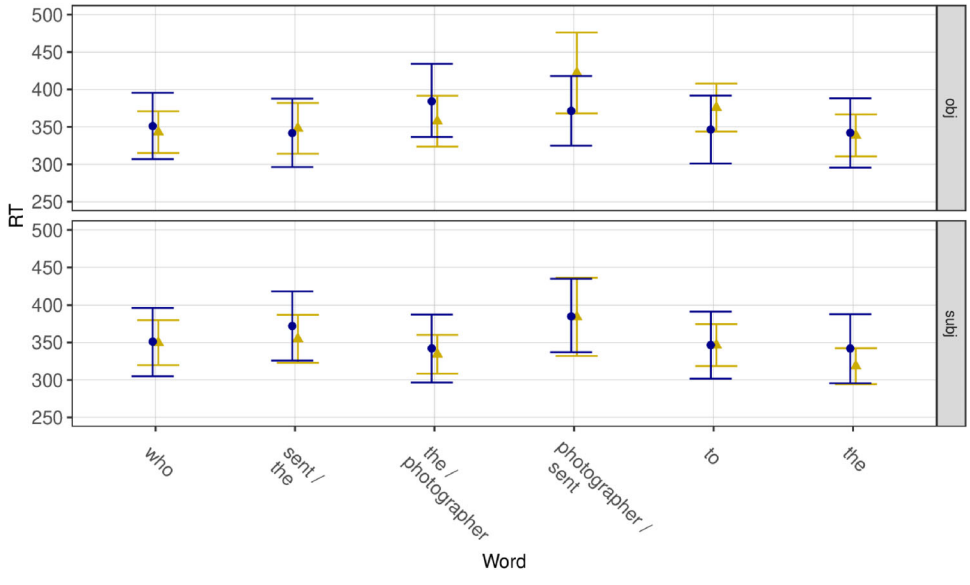


Fig. 11. Model 3 of reading—posterior predictive. The blue dots are predicted mean RTs and the blue bars provide the 95% credible intervals. The observed data are in yellow. The yellow triangles are observed mean RTs, and the yellow bars are +/− 2 standard errors, taken from Grodner and Gibson (2005).

be seen when one compares the size of 95% credible intervals of this model and Model 1, see Fig. 9. Indeed, the median of *SD*, the parameter for the standard deviation of the likelihood, is estimated at 31, twice as large compared to Model 1.

When we turn to Model 3, Fig. 11, we see again a worse fit in object-relative clauses. The verb in object-relative clauses is processed too quickly according to the model (the credible interval does not include the actual mean), arguably because no syntactic processes related to gap resolution slow down the reader. As was the case in Model 2, the 95% credible intervals include most mean RTs but at the price of being less precise about posterior distributions of reading times. This can be seen from the size of 95% credible intervals of Model 3 compared to Model 1 and from the fact that the median of *SD* is 25. Since the estimated parameters are tied to fewer processes in each word (F, f only affect the lexical retrieval), we see that even the best estimation of these parameters does not suffice to correctly predict the data—more is needed than just lexical retrieval.

The best predictive accuracy of Model 1 is also clearly visible from its lowest widely available information criterion (WAIC) (Gelman et al., 2013): $WAIC(\text{Model 1}) = 102.0(SE = 3.0)$, $WAIC(\text{Model 2}) = 119.0(SE = 6.4)$, $WAIC(\text{Model 3}) = 113.9(SE = 6.4)$.

4.1.7. Summary

The presented case study modeled the self-paced reading experiment from Grodner and Gibson (2005). It showed that the symbolic predictions of the data-driven cue-based model of parsing are in agreement with reading data. It also showed that it is possible to develop an end-to-end model, which carries out the reading task just as participants of Grodner and Gibson (2005) had to do, and in which an estimation of four ACT-R parameters for the whole model is sufficient to fit observed mean RTs. This provides evidence that the cue-based model of parsing can be combined with other cognitive processes to simulate data from an experimental task like self-paced reading.

4.2. Case 2: Lexical and syntactic processing

In the first case study, we focused on the interaction between the retrieval of parsing steps and the retrieval of dependency. The second case study focuses on the interaction between the retrieval of parsing steps and the lexical retrieval.

The interaction between lexical processing and syntactic processing has been investigated in the model of eye control in reading, E-Z Reader (in particular, E-Z Reader 10; see Reichle, Rayner, & Pollatsek, 2003; Reichle, Warren, & Mcconnell, 2009). E-Z Reader proposes a so-called staged architecture: the lexical process and the syntactic process are sequentially ordered; lexical processing precedes integration, which syntactic processing is part of. E-Z Reader provides a subsymbolic system that integrates the staged architecture assumption and allows psycholinguists to develop quantitative predictions for eye-tracking data.

A disadvantage of E-Z Reader is that it leaves it unclear how symbolic systems translate to the subsymbolic equations. This is of less concern for lexical processing; however, symbolic processes like syntactic parsing cannot be straightforwardly linked to E-Z Reader calculations.

The cue-based model of parsing does not face this challenge. We have seen that parsing steps can be translated into a quantitative measure (activations), and we have seen that this measure can be translated into reading times. Moreover, this translation is not postulated ad hoc. It is not created for this case of retrieval or just for this model. Rather it builds on the independent ACT-R findings. Thus, there is a possibility that the cue-based model of parsing can advance the impressive research on eye control in reading developed by the E-Z Reader community.

I will use the third eye-tracking experiment of Staub (2011) to study whether the model can simulate lexical and syntactic processing and the interaction thereof. The experiment consisted of 2 × 2 conditions, summarized in (25). There were two manipulations: (i) in the critical region, italicized in (25), either a high-frequency word (*walked*) or a low-frequency word (*ambled*) was used; and (ii) the same word could either be integrated with the previous words (the Grammatical condition) or it could not be integrated (the Ungrammatical condition). In the example, the ungrammaticality is driven by the fact that the preceding word was a preposition which in this sentence cannot be followed by an *-ed* word.

(25)

a. The professor saw the students that *walked* across the quad.
(Grammatical, High Frequency)

b. The professor saw the students that *ambled* across the quad.
(Grammatical, Low Frequency)

c. The professor saw the students over *walked* across the quad.
(Ungrammatical, High Frequency)

d. The professor saw the students over *ambled* across the quad.
(Ungrammatical, Low Frequency)

Three ROIs were measured in Staub (2011): the pre-critical word (*that* or *over* in (25)), the critical word (*walked* or *ambled* in (25)), and the spillover, the three words following the critical word (*across the quad*.). Of the standard eye-tracking measures, I will focus on first-pass reading times and regressions, which revealed the effect of lexical and syntactic manipulations (see Staub, 2011 for details).

In Section 4.2.1, we consider the structure of the model with eye control. In Section 4.2.2, we look at the structure of the Bayesian model for free parameters. In Section 4.2.3, the results of the model are discussed.

4.2.1. Sequential model for natural reading

The model is almost identical to the model used for Grodner and Gibson (2005), see Section 4.1.1. There are two differences: no motor module for controlling key presses is involved since we do not model self-paced reading but eye tracking; second, we now have to be explicit about the eye control module.

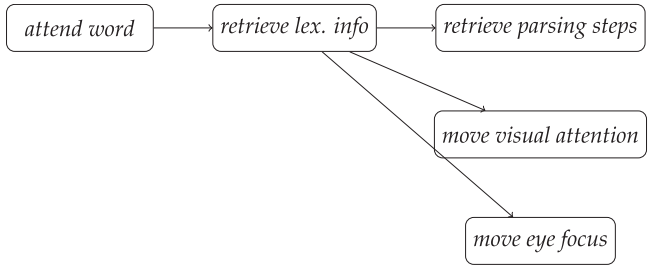


Fig. 12. Sequential model of reading on one word for eye-tracking simulation. Each box represents one subprocess arrows the order. When arrows branch, this signals parallel processing, that is, two processes running concurrently.

The scaffolding of the eye control module is taken over from E-Z Reader. Our model is built on EMMA, which generalizes and simplifies the assumptions of E-Z Reader (Salvucci, 2001). Its general structure is shown in Fig. 12. This structure is compatible with the sequential model used for self-paced reading, see Fig. 4.11 What is important to observe is that linguistic processing is split into the lexical processing and syntactic processing and the two parts are interspersed with eye-movement/attention control: the attention and eye movements are programmed to move after the lexical retrieval is finished and at the same time that syntactic processing starts. This is largely similar to the position of E-Z Reader, with one simplification: E-Z Reader postulates another lexical access after eye movement to the next word was programmed.

Just as in E-Z Reader, eye movement control is split into two stages: the initiation phase, in which eye saccade is planned; the execution of the movement. And just as in E-Z Reader, it is assumed that independently of eye movement control, visual attention is organized. The attention moves to the next word at the same moment as the eye movement is programmed and the move is instantaneous. However, attending to an object takes more time when the object is further away from the eye focus. The visual encoding time is calculated as shown in (26). d is the distance between the object and the current eye position, measured in degrees of visual angle. D is the visual properties of the object. Following Dotlačil (2018), I take D to correspond to word length, measured in the number of characters. For more details on EMMA and E-Z Reader, see Salvucci (2001) and Staub (2011).

$$\text{Visual encoding } T_{enc} = K \cdot D \cdot e^{kd} \qquad (K, k \text{ - free parameters}) \tag{26}$$

The lexical processing is the same as for the previous model in Section 4.1. The syntactic processing is almost identical. As was the case for model in Section 4.1, we calculate retrieval times from the average activation of parsing steps.

Two modifications are added and they both are connected to the fact that the model will now simulate regressions, not just reading times. It is assumed that regression takes place in those two cases from word n:

- when the activation of a parsing step is below a retrieval threshold, t, the parsing step on word n is not retrieved and eyes are programmed to regress;

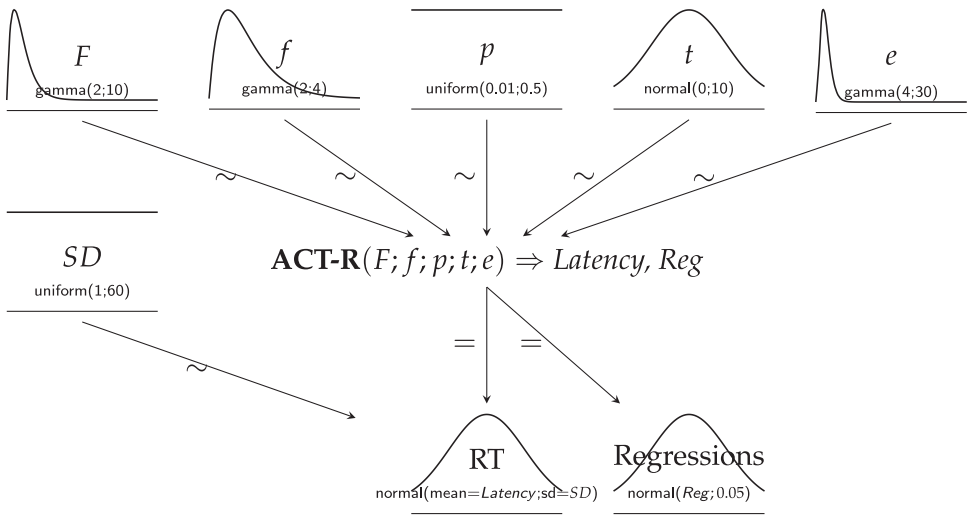


Fig. 13. Bayesian model for parameter estimation of Staub (2011).

- when, on a word n, a reanalysis takes place (i.e., the syntactic analysis of n is not compatible with the analysis proposed on the word $n-1$, see also Section 4.1.3), the regression is triggered with the probability p.

These cases signal that the word n cannot be straightforwardly incorporated either because no parsing step can be recalled (case 1) or because a reanalysis is triggered (case 2).

The regression interacts with eye control just as in E-Z Reader. It launches from the word that the eye focus is on, unless the eye movement control is in the non-labile saccade phase—in that case, regressions wait for the end of execution and are triggered at the next word.

4.2.2. Bayesian modeling

Five ACT-R parameters are modeled. There are two parameters affecting the (lexical and syntactic) retrieval: F, f. Two parameters model regressions: t (the threshold) and p (the probability of regression due to reanalysis). One parameter controls eye movements: e, the amount of time it takes to prepare an eye shift. The other parameters are kept at their default values (as was the case for the previous model) and r is kept at the median value observed in the previous study (33 ms).12 Five parameters might seem like a lot, but keep in mind that we develop a cognitive model, we are not trying to fit the data to a regression model. This means that the parameters are kept the same across all three regions and both measures (24 data points in total) and the cognitive model has to model the whole process of reading with (just) these parameters.

The structure of the Bayesian model is shown in Fig. 13. As was the case with the previous model, the model runs all stimuli from the experiment, collects all reaction times and regressions, and compares that the actual mean first pass reading times and mean probability of regressions. Apart from the five ACT-R parameters, we also model the *SD* parameter, the

standard deviation of normal distribution that models the likelihood for the RT data (see also Section 4.1.4).

The following prior structure for the parameters is assumed:

- $F \sim Gamma(\alpha = 2, \beta = 10)$
- $f \sim Gamma(\alpha = 2, \beta = 4)$
- $p \sim Uniform(0.01, 0.5)$
- $t \sim Normal(0, 10)$
- $e \sim Gamma(4, 30)$
- $SD \sim Uniform(1, 60)$

The ACT-R parameters that were modeled in Section 4.1 have the same priors as the previous model (see Section 4.1.4 for justification).

Let us turn to free parameters that are unique to this model, starting with p. We have very little evidence for any value of p, the probability of regression, apart from the fact that it cannot be an extremely large value, given that the highest mean of the probability of regression is 0.59. So, we keep the prior uniform and assume that it cannot be higher than 0.5, slightly lower than the highest mean (keep in mind that there are two ways to trigger regressions and p plays a role only in one of them). The threshold t is by default set at 0 (measured on the activation scale). We assume the prior to be a normal distribution with mean 0 and sd 10. This is a very broad, unrestricted prior since no recalled elements have an activation smaller than -10 and greater than $+10$. Finally, the prior of e is a gamma distribution, whose mean is the default value. e is measured in seconds. We assign most weight to values between 0 and 0.2—it seems very plausible that eye movement preparation should not be larger than 0.2 s (200 ms).

The estimation is done using PYMC3 and MCMC-sampling with 3,000 draws, 2 chains, and 200 burn-in. The Rhat values of the samples for all the parameters were lower than 1.05.

4.2.3. Results

The results for first-pass times are summarized in Fig. 14. The triangles indicate the observed mean first pass reading times for the pre-critical, critical, and spillover region, the segments provide the 95% CRIs (credible intervals) for the first pass reading times predicted by the ACT-R model using the posterior distribution of the ACT-R parameters, and the dots are the predicted mean first pass reading times.

Two things are worth observing. First, the model is able to predict first pass reading times per region: the pre-critical region is the fastest, the critical region is in between, and the spillover region is the slowest. This model correctly derives this behavior even though there is no "intercept" or "region" condition in the model—all the measures have to fall out from its simulation of reading. Second, the model, correctly and in accordance with E-Z Reader, generates increased reading times on the critical word as a factor of frequency, not grammaticality. The effect of frequency is washed away in the spillover region, largely in accordance with the data (there is a small effect of interaction between frequency and grammaticality, which cannot be modeled by the current model and which is reported as non-significant in Staub (2011)).

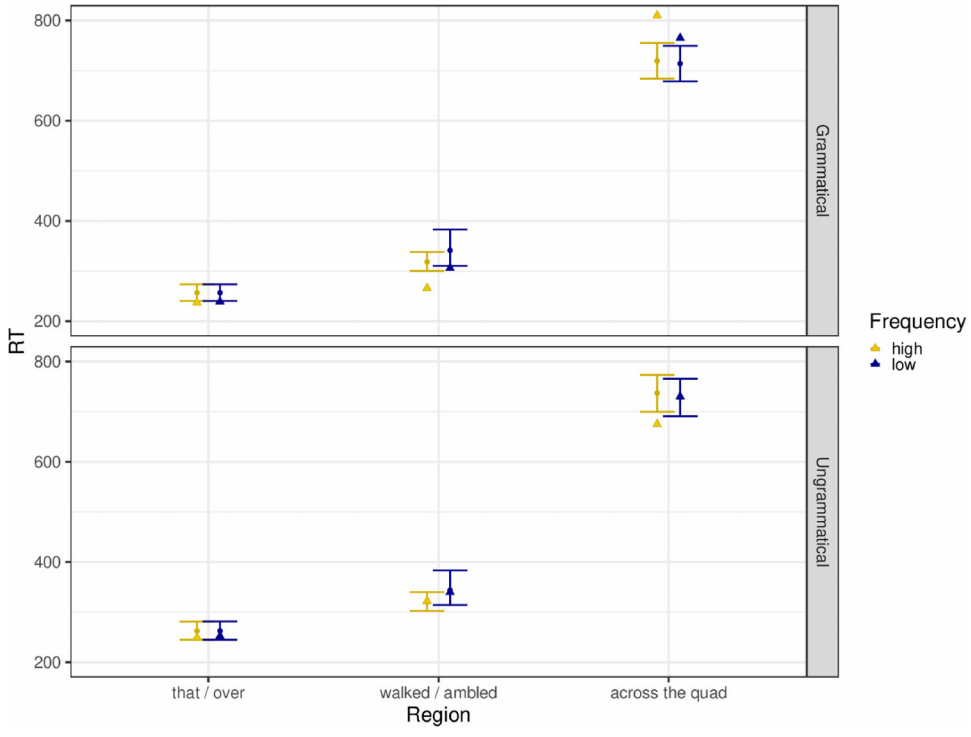


Fig. 14. First pass reading times—predictions and data. The dots are predicted mean first pass reading times. The bars provide the 95% credible intervals. The predictions come from the ACT-R model using the posterior distribution of the parameters. The triangles are observed first-pass reading times, taken from Staub (2011).

Let us look at the model of regressions, which is driven by the syntactic processing. Before we turn to details, I will make some general observations. The data in Staub (2011) show that there are more regressions in the critical region in the ungrammatical condition compared to the grammatical condition. How could the cue-based model of parsing simulate that? There are two possibilities. First, it would fall out from our model if the activations for the ungrammatical sentences were smaller than the activations for the grammatical sentences when parsing the critical word. Second, this could happen if the ungrammatical condition triggered reanalyses more often.

Let us start with the first one and reason about why we observe it. Transition-based shift reduce parsers are quite robust, in the sense that they seldom halt (McDonald & Nivre, 2011). However, there is one difference between ungrammatical/hard-to-parse sentences and grammatical ones. In the ungrammatical case, the declarative memory will not carry chunks that will match as many cues in the current context as in the case of grammatical sentences. After all, since we are dealing with an ungrammatical sentence, we are building a structure that has most likely not been observed before. Since ungrammatical sentences will find chunks that match the current context in fewer cues, they will spread activation less, and consequently,

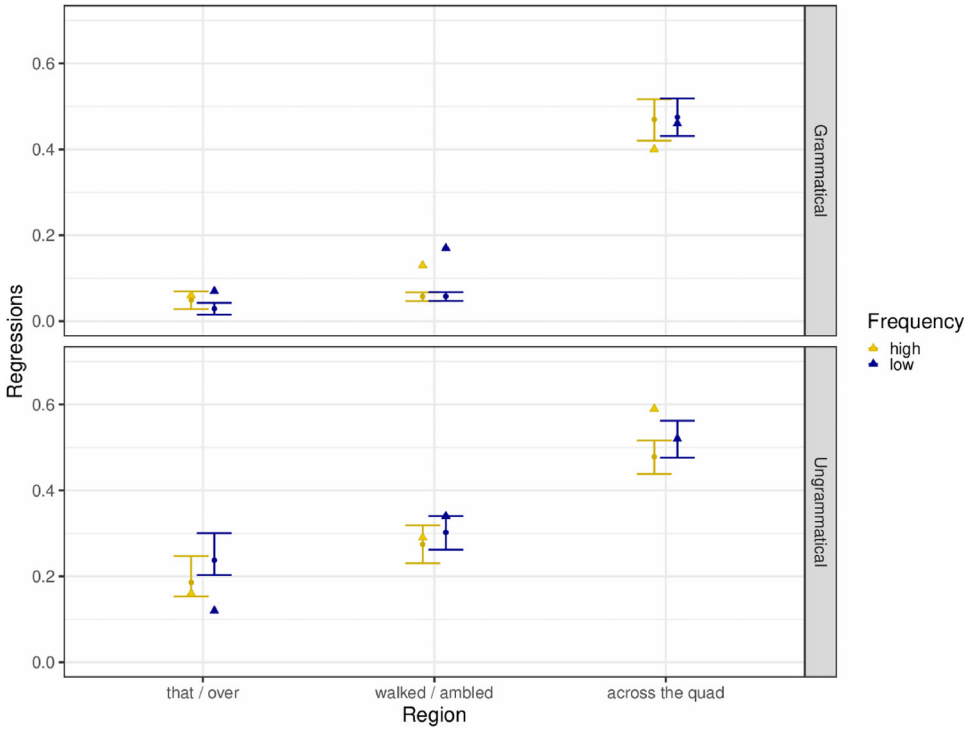


Fig. 15. Regressions—predictions and data. The dots are predicted probabilities of regressions. The bars provide the 95% credible intervals. The predictions come from the ACT-R model using the posterior distribution of the parameters. The triangles are observed probabilities of regressions, taken from Staub (2011).

the activation of the retrieved chunks will be lower than the highest activation of the chunks retrieved for grammatical sentences.

We also observe the reanalysis in ungrammatical sentences because the grammatical parse proposed up to then turned out to be incorrect.

Due to both reasons, we expect that we should observe increased regressions in the ungrammatical sentences on the point at which the ungrammaticality is triggered.

Let us now check the quality of the quantitative fit. The results are shown in Fig. 15. The mean regressions are largely captured correctly. The data-driven model definitely correctly captures the contrasts between grammatical and ungrammatical conditions. However, there is room for improvement. The model underestimates regressions in the critical region (Region 2) in the grammatical condition, as if it expected the grammaticality effect to be larger than actually observed.

Apart from the effect on the critical word, the model also predicts that the grammaticality will affect regressions on the pre-critical word. This is in accordance with the data but was not predicted by Staub (2011) and E-Z Reader. The cue-based model of parsing correctly predicts this contrast because it happens so that the (27-a) has a higher activation of parsing steps at

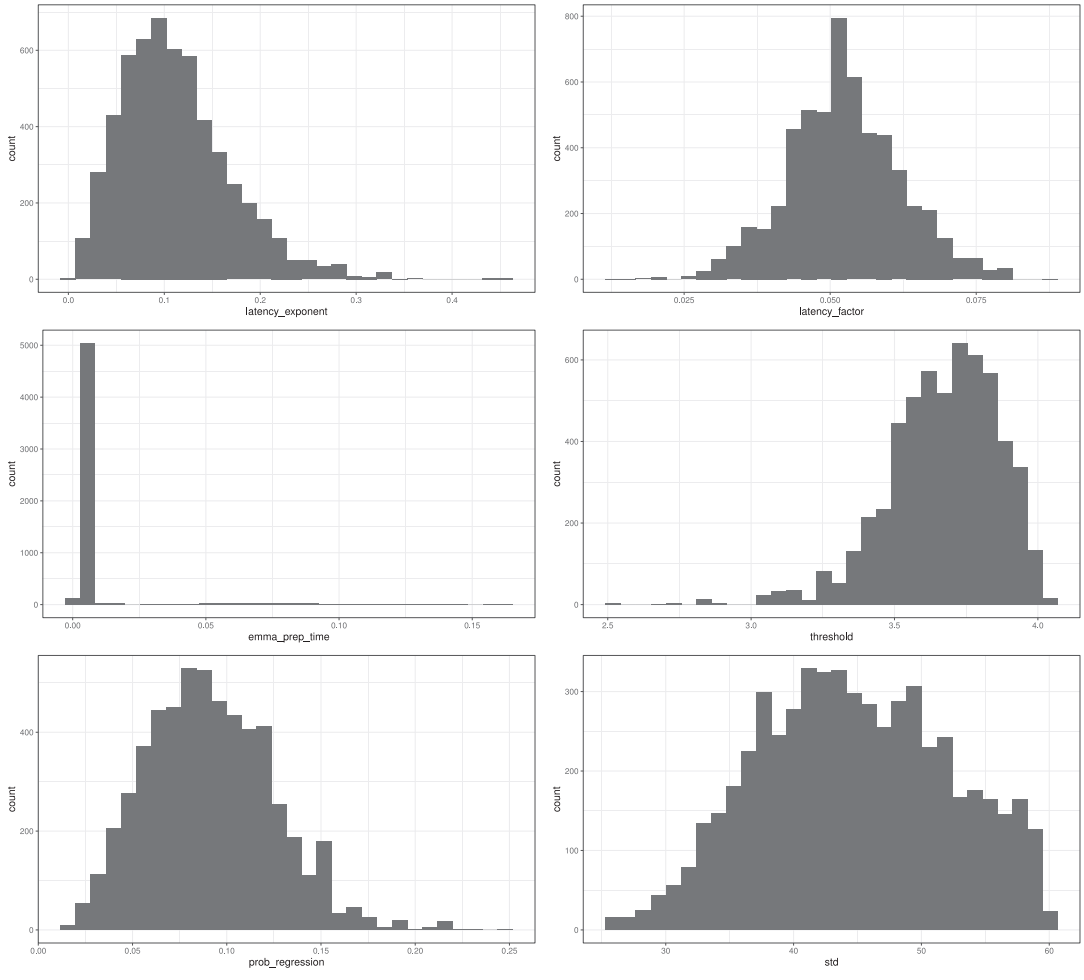


Fig. 16. Posteriors for the six parameters estimated in the Bayesian ACT-R model.

the pre-critical word compared to (27-b).

a. The professor saw the students that . . .

b. The professor saw the students over . . . (27)

The estimated values of the parameters are summarized below. See also Fig. 16.

- F—median: 0.05, sd: 0.01
- f—median: 0.10, sd: 0.06
- e—median: 0.006, sd: 0.02
- p—median: 0.09, sd: 0.03
- t—median: 3.7, sd: 0.2

- *SD*—median: 44.7, sd: 7.4

The first two parameters have also been estimated in the first case study. The medians for both parameters, F and f, are within one standard deviation of the median values found in Case study 1 and the posterior distributions are similar (cf. Figs. 8 and 16). The convergence is very encouraging.

p and t are hard to interpret on their own.13 e, the preparation phase for eye movement, is at 0.006 s. This is a very low value and as far as I can see, the most worrisome issue with the model that future research could improve upon since the preparation phase for eye movement is commonly taken to be much greater (around 100 ms). The low value is likely caused by the fact that parafoveal attention is limited in the current model, so eyes have to move rather quickly to attend to upcoming words. The challenge is that increasing parafoveal attention results in the model skipping words, which significantly increases the complexity of Bayesian modeling (see also footnote).

4.2.4. Summary

The second case study modeled the eye-tracking reading experiment from Staub (2011). We saw that the ACT-R architecture allows us to build an E-Z Reader style model for eye control and eye movement that interacts with language comprehension. It was shown that the cue-based model of parsing links symbolic properties of the parser to subsymbolic values and generates detailed quantitative predictions for eye movements that are to a large extent correct. This provides evidence that the cue-based model of parsing can be combined with lexical retrieval and eye control to simulate data from an experimental task like eye-tracking reading.

4.3. Case 3: Modeling corpus data

So far, we saw that the syntactic parser constructed as a cue-based retrieval can to a large extent correctly match reading data from individual experiments. We now go beyond selected experiments and show that the model generalizes to a larger pool of data.

In the current section, we will look at the predictions of the model for the Natural Stories corpus (NSC, Futrell et al., 2018). The Natural Stories corpus is a corpus containing 10 English narrative texts with 10,245 lexical tokens in total. The texts were edited to contain various syntactic constructions, including constructions that are very rare. The corpus was read by 181 English speakers using a self-paced reading moving-window paradigm and the self-paced reading data were released along with the texts. Furthermore, all the sentences were annotated according to Penn Treebank notational conventions by the Stanford Parser (Klein & Manning, 2003) and hand-corrected. The fact that the NSC has a plethora of syntactic constructions and includes manually controlled PTB-compatible syntactic parses makes the corpus particularly suitable for our goal.

4.3.1. Parsing model

Unlike in the previous cases, we do not try to model all the processes in reading to fit the model as closely as possible to reading time data. We only observe whether the syntactic processing model is a good predictor for reading data.

Table 1
The intercept and ACTIVATION slope estimates with corresponding t-values for a mixed effect model $log(RT) \sim 1 + \text{ACTIVATION} + (1 + \text{ACTIVATION}|\text{SUBJECT}) + (1 + \text{ACTIVATION}|\text{TEXT})$

	Estimate	t-Value
INTERCEPT	5.7	273
Activation	**−0.008**	**−6.1**

We proceed as follows. Per word, we collect the average activation of retrieved parsing steps from the same declarative memory model used in the previous case studies. Since the NSC is self-paced reading corpus data, the parsing model assumes self-paced reading, that is, at the moment of retrieving parsing steps, it cannot look ahead and collect information about the upcoming words. We expect that the level of activation should negatively correlate with reading times (see Section 4.2). This finding would strengthen the evidence for the cue-based model of parsing.

One worry might be that the syntactic processing might go astray, even more so because the NSC uses infrequent syntactic constructions. To avoid this, we collect at every word the correct syntactic parse at that word, as provided by the NSC. This correct parse is used as the context for retrieval: based on this parse, the retrieval of parsing steps is attempted and the average activation of the retrieval is recorded. That means that the parser will have the correct syntactic structure at every word and will use that context for retrieval. Thus, we can be sure that whatever we are to find, the finding is not obfuscated by the fact that the parser built an incorrect cognitive context that it uses for cue-based retrieval.

4.3.2. Results

The results are summarized using mixed-effects models with the dependent variable log-transformed reading time (logRT) and random factors subject ($n = 181$) and text ($n = 10$). We start with a simple model with just one fixed effect, ACTIVATION (z-transformed), the averaged activation of retrieved parsing steps per word, and by-subject ($n = 181$) and by-text ($n = 10$) random intercept and random ACTIVATION slope. The results are summarized in Table 1. The model shows that the effect of ACTIVATION is significant and goes in the expected directions: higher activations of retrieved parsing steps correspond to a decrease in logRTs.

A more complex model in which various low-level confounding factors are included is also considered. The following confounds are taken into account: (i) POSITION (the word position in a sentence, z-transformed), (ii) ZONE (the word position in the whole text, z-transformed), (iii) WORD LENGTH (the length of the word as the number of characters, z-transformed), (iv) LOG(FREQ) (log-unigram frequency), (v) the interaction of word length × log unigram frequency, (vi) LOG(BIGRAM) (log bigram probability), and (vii) LOG(TRIGRAM) (log trigram probability). Frequencies, bigram and trigram probabilities are provided in the NSC. In the model, we ignore the first of each sentence since the first words are often outliers, and furthermore, bigram and trigram probabilities cannot be calculated at the beginning of a sentence. We also ignore words directly followed by punctuation marks since these are known to show wrap-up effects, not modeled by the parser. Finally, the model included by-subject

Table 2
Estimates for the mixed effect model $log(RT) \sim 1 + \text{POSITION} + \text{ZONE} + \text{WORD LENGTH} * \text{LOG(FREQ)} + \text{LOG(BIGRAM)} + \text{LOG(TRIGRAM)} + \text{ACTIVATION} + (1 + \text{ZONE} + \text{ACTIVATION}|\text{SUBJECT}) + (1 + \text{ACTIVATION}|\text{TEXT})$

	Estimate	*t*-Value
INTERCEPT	5.71	266.62
POSITION	-9×10^{-6}	−0.02
ZONE	−0.044	−14.92
WORD LENGTH	0.062	22.53
LOG(FREQ)	−0.002	−5.4
WORD LENGTH:LOG(FREQ)	−0.003	−19.3
LOG(BIGRAM)	−0.0006	−1.8
LOG(TRIGRAM)	−0.002	−6.3
Activation	**−0.003**	**−3.3**

random intercept and random ACTIVATION and ZONE slopes and by-text random intercept and random ACTIVATION slope.14 The results of the model are shown in Table 2. We see that after adding the confounds, ACTIVATION importantly remains significant and the effect goes in the expected direction, showing that the role of activation of parsing steps cannot be explained (away) by the considered low-level factors.

We now proceed to another model, which breaks down the role of ACTIVATION and can reveal what drives the effect observed in Table 2. Two possibilities for the source of the ACTIVATION effect are of theoretical interest. We know that the activation *increases* with the increase in the number of cues that match between the context and the retrieved chunks (i.e., the facilitatory effect of partial distractor match in ungrammatical sentences, see Section 2.2). It is possible that our finding in Table 2 is driven by the number of cues matching, that is, the increase in the matching features correlates with a *decrease* in logRTs. We also know that the activation *decreases* with the size of fan of cues: if a cue matches many parsing steps, it is not very useful and does not boost activation as much as when it matches only few parsing steps (i.e., the inhibitory interference due to partial distractor match in grammatical sentences, see Section 2.2). If the fan size was the driving force, we would expect that the increase in fan size correlates with an *increase* in logRTs.

To investigate this, we consider the model whose estimates and *t*-values are summarized in Table 3. In this model, we substitute ACTIVATION with the z-transformed factors # MATCHING CUES (how many cues are matching?) and FAN SIZE (the average fan size of cues). The effect of # MATCHING CUES is highly significant and goes in the expected direction. The effect of FAN SIZE is non-significant. We can conclude that the effect of activation is driven by the match in features, rather than the size of the fan of labels. It is left open to the future research why the number of matching features, but not the fan size, seems to be crucial in modeling reading times and the effect of parsing on reading times, at least in the case of the Natural Stories Corpus.

Next, we investigate the question of how the observed effect of activation on reading times compares to well-investigated and related theoretical concepts in computational psycholinguistics: surprisal from Surprisal Theory (Hale, 2001) and integration cost from Dependency

Table 3
Estimates for the mixed effect model $log(RT) \sim 1 + \text{POSITION} + \text{ZONE} + \text{WORD LENGTH} * \text{LOG(FREQ)} + \text{LOG(BIGRAM)} + \text{LOG(TRIGRAM)} + \#\ \text{MATCHING CUES} + \text{FAN SIZE} + (1 + \text{ZONE} + \#\ \text{MATCHING CUES}|\text{SUBJECT}) + (1 + \#\ \text{MATCHING CUES}|\text{TEXT})$

	Estimate	*t*-Value
INTERCEPT	5.7	267.4
POSITION	−0.0001	−0.4
ZONE	−0.044	−14.9
WORD LENGTH	0.059	21.3
LOG(FREQ)	−0.002	−7.4
WORD LENGTH:LOG(FREQ)	−0.003	−18.0
LOG(BIGRAM)	−0.0006	−1.8
LOG(TRIGRAM)	−0.002	−6.2
# Matching cues	**−0.004**	**−7.2**
Fan size	**−0.00008**	**−0.18**

Locality Theory (Gibson, 1998, 2000) (see also Section 5 for a comparison of the cue-based model of parsing to Surprisal theory and other related works). We consider a model in which, besides activation and the low-level factors introduced above (see Table 2), the following measures from computational psycholinguistics are added: a surface surprisal estimate, namely, 5-gram surprisal trained on Gigaword corpus (Graff & Finch, 2007), two hierarchical surprisal estimates, namely, a surprisal using the parser from Van Schijndel and Schuler (2013) trained on the Penn Treebank data sections 2 through section 21 reannotated using generalized categorial grammar, GCG (Nguyen, Schijndel & Schuler, 2012) and a probabilistic context-free grammar (PCFG) surprisal trained on the same sections of Penn Treebank but using the original labels, that is, no reannotation, and finally, integration cost of DLT that additionally assumes that coordination is less expensive and that excludes modifier dependencies. Finally, since it is possible that the effect of just introduced psycholinguistic measures spills over to the following words (see also Shain & Schuler, 2019 for detailed investigations of spillover effects), the model also includes one-word spillover for each of the five predictors. The surprisal values (apart from the PTB with no reannotation) were also used in Van Schijndel and Schuler (2015); Shain, Blank, Schijndel, Schuler, and Fedorenko (2020) and the DLT values were also used in Shain et al. (2016).15 The model summary is given in Table 4. ACTIVATION remains significant even after these psycholinguistic measures are added.

Finally, Table 5 summarizes log-likelihood of the models that use the same low-level factors as Table 2 plus one of the following theoretical measures: our main measure of interest, activation, the averaged activation of retrieved parsing steps per word (line 1), 5-gram surprisal (line 2), PCFG surprisal using reannotated generalized categorial grammar, GCG (line 3), PCFG surprisal using the original PTB annotation (line 4), and DLT that assumes that coordination is less expensive and that excludes modifier dependencies (line 5). Every model in Table 5 also has by-subject random intercept and random ZONE slope and by-text random intercept. The 5-gram surprisal turns out to be the best model. The model that collects activations is worse than the surface estimate of surprisal (line 2) and the surprisal estimate based

Table 4
Estimates for the mixed effect model $log(RT) \sim 1 + \text{Position} + \text{Zone} + \text{Word Length} * \text{Log(freq)} + \text{log(bigram)} + \text{log(trigram)} + \text{5-gram surprisal} + \text{5-gram surprisal spillover} + \text{GCG surprisal} + \text{GCG surprisal spillover} + \text{PTB surprisal} + \text{PTB surprisal spillover} + \text{DLT} + \text{DLT spillover} + (1 + \text{Zone} + \text{5-gram surprisal} + \text{GCG surprisal} + \text{activation}|\text{Subject}) + (1 + \text{zone}|\text{Text})$

	Estimate	*t*-Value
Intercept	5.7	264
Position	0.001	2.8
Zone	−0.045	−15.1
Word length	0.06	21.3
Log(freq)	−0.0002	−0.4
Word length:Log(freq)	−0.003	−17.9
Log(bigram)	−0.0004	−1.0
Log(trigram)	−0.0002	−0.6
5-gram surprisal	0.01	7.7
5-gram surprisal spillover	0.01	17.9
GCG surprisal	0.007	5.5
GCG surprisal spillover	0.006	4.3
PTB surprisal	−0.002	−1.7
PTB surprisal spillover	0.003	3.0
DLT	−0.002	−4.7
DLT spillover	−0.0004	−1.0
Activation	**−0.002**	**−2.8**

Table 5
Log-likelihood comparison of a model with activations to surprisal models and a model with DLT integration cost

Measure	log-Likelihood
Activation	**−76,855**
5-gram surprisal	−76,748
surprisal with GCG annotation of PTB	− 76,790
surprisal with original PTB annotation	−76,867
DLT	−76,890

on PTB with GCG reannotations (line 3). However, our model with activations has a better fit to data compared to the model with surprisal based on the original PTB annotations (line 4) and to the model based on DLT (line 5). Of the comparisons, the comparison between our model and the model with surprisal based on the original PTB annotations is arguably the most important since these two models were trained on the same data set using the same (PTB) labels. In this case, the cue-based model of parsing compares favorably to surprisal.

4.4. Summary of the results

We have inspected three case studies that tested the predictions of the cue-based model of parsing:

- Case study 1 simulated the self-paced reading experiment of Grodner and Gibson (2005). The study shows that it is possible to construct a good-fitting reading model in which lexical retrieval, dependency retrieval and parsing are built based on the same memory structures restricted by the same parameter values.
- Case study 2 simulated the eye-tracking experiment of Staub (2011). The study shows that the cue-based model of parsing can provide a link between the symbolic system (parses) and behavioral measures (reading times and regressions). It also shows that it is possible to build one model in which lexical retrieval and cue-based retrieval of parsing interact in the E-Z Reader style with eye control and in which all parsing and lexical retrieval are built based on the same memory structures restricted by the same parameter values.
- Case study 3 correlated the measure of parsing step availability, stored in activations, with reading times using the data from a self-paced reading corpus, the Natural Stories Corpus (Futrell et al., 2018). The study shows that the cue-based model of parsing is a significant predictor of reading times, even after various possible confounds are considered and after estimates of other measures often used in psycholinguistics, surprisal, and integration cost in DLT are included. The prediction is driven by the matching cues between the context and the retrieved parsing step.

The Case studies 1–3 provide evidence for the cue-based model as a computational model of human parsing.

5. Comparison to related works

In this section, it is discussed how the cue-based model of parsing compares to related proposals of parsing in computational (psycho)linguistics, including ACT-R linguistic models.

5.1. Surprisal

At least since Hale (2001), it has become very common to model processing difficulties using quantitative distributions estimated on other data, as is the standard procedure in computational linguistics. This paper follows this methodological line.

Arguably, the dominant method to study the impact of parsing on online behavioral measures is to use the theory that connects processing difficulties to the surprisal of a word given its syntactic context, as introduced in Hale (2001) (see also Levy, 2008). This account is commonly labeled Surprisal theory. The theory has been supported by corpus investigations (Boston, Hale, Vasishth, & Kliegl, 2011; Demberg & Keller, 2008). It has also been validated in controlled experiments (Jäger, Chen, Li, Lin, & Vasishth, 2015; Levy, Fedorenko, & Gibson, 2013; Linzen & Jaeger, 2016; Wu, Kaiser, & Vasishth, 2017) even though see also Vasishth, Mertzen, Jäger, and Gelman (2018) for a failure to replicate the evidence for surprisal reported in Levy et al. (2013). This section does not focus on empirical issues with Surprisal theory, though. Rather, the goal is to compare the theory to the current approach. As

we will see, the two approaches share several assumptions about the bottleneck that causes processing difficulties.

Surprisal theory states that processing difficulties are related to the self-information (also known as surprisal) of the event that the word w_n occurs, given the preceding context *ctxt*. In syntactic analyses, the preceding context *ctxt* is treated as equivalent to the words appearing prior to w_n in the same sentence, that is, $w_1 \ldots w_{n-1}$.

$$-\log(p(w_n|ctxt)) \tag{28}$$

Levy (2008) shows that under some reasonable restrictions, (28) is equivalent to the Kullback-Leibler (KL) divergence (also known as a relative entropy), see (29), where q is the probability distribution over structures given *ctxt* and p is the probability distribution over structures (T) given *ctxt* and w_n.

$$D_{KL}(p||q) = \sum_T p(T) \log \frac{p(T)}{q(T)} \tag{29}$$

The equivalence plays a role in the interpretation of Surprisal theory. One can think of Surprisal theory as an account that links processing difficulties to a high relative entropy between p and q. According to this interpretation, we can assume that readers track probability distributions over structures during incremental interpretation and when p, the probability distribution over structures given *ctxt* and w_n, strongly diverges from q, the distribution over structures given just *ctxt*, processing cost is incurred.

The formulas in (28) and (29) are also closely related to the cue-based model of parsing, as we will see now.

Recall that association strength in ACT-R is the pointwise mutual information between cues and the chunk i, see (30), where p is a probability function, i is a chunk, and c is a cue in the current context. One can think of the chunk that i is a parsing step needed to be recalled to integrate w_n.

$$\log \frac{p(i, c)}{p(i)p(c)} \tag{30}$$

Spreading activation is the expected value of the pointwise mutual information (also known as mutual information) and is calculated for a single chunk as shown in (31).16

$$\sum_{c \in cues} p(c) \log \frac{p(i, c)}{p(i)p(c)} \tag{31}$$

Mutual information and the KL divergence are closely related. Using the relation between the two information-theoretic notions, we can rewrite the last formula as follows:

$$D_{KL}(p(i, cues)||p(i)p(cues)) \tag{32}$$

To generalize the last formula, let us think of cues as the context preceding the word w_n (as is done in ACT-R, where cues represent the current cognitive context of the agent; see also Section 2.2). The spreading activation measures how different the joint distribution of the parsing step and the context is from treating the parsing step and the context as independent.

A high divergence signals that the parsing step and the context are dependent, a divergence close to 0 signals that they are independent. Since the additive inverse of activation is used to calculate observable difficulties like increased retrieval times and increased chances of retrieval failure, it is predicted that the more the parsing step and the current cognitive context are independent of each other, the more observable processing difficulties there are.17

We thought of i as a parsing step and *cues* as a cognitive context, because this was the implementation of mutual information in this paper. However, this is not the only possible implementation. Generalizing to any structures gives us (33), where T are the structures generating *ctxt* and w_n, while C are all the structures generating *ctxt*. The cue-based model of parsing is a particular implementation of (33).

$$D_{KL}(p(T, C)||P(T)P(C)) \tag{33}$$

The point of difference between (33), the relative-entropy interpretation of cue-based model, and 29, the relative-entropy interpretation of surprisal, is that instead of measuring the divergence between two probabilities over structures, we measure the divergence between their joint distribution and their independence.

While both interpretations of processing difficulties seem plausible, there is a difference between Surprisal theory and the cue-based model of parsing. The former is established to account for parsing effects. The latter, however, is built inside a cognitive architecture. Treating spreading activations according to (32) is motivated independently of parsing. The motivation comes from other linguistic studies (Lewis, Vasishth, & Van Dyke, 2006) and a wide range of data on human cognition (Anderson & Lebiere, 1998; Anderson & Reder, 1999; Anderson et al., 2004). Consequently, when the cue-based model of parsing has to be fit to behavioral measures, modelers do not have the freedom to fit parsing independently of other cases of retrieval: every recall is treated the same way. Another way to look at this is that the approach in this paper provides a single model (ACT-R retrieval) to explain processing difficulties caused by expectations given the constructed syntactic context and difficulties due to the recall of recently constructed dependents (such as wh-elements in relative clauses). Finally, embedding the model in a general cognitive architecture allows researchers to principally connect the theory to observable behavioral data. Indeed, to the extent the fit to behavioral data in Section 4.1 and Section 4.2 can be seen as success, we have evidence that the cue-based model of parsing is well positioned to not only predict processing difficulties, but also to model reading times and regressions using one and the same model for any type of retrieval. This is in contrast to previous traditions that commonly treat memory-based processing difficulties and difficulties due to expectations given the syntactic context as separate even in models that try to investigate their joint effect on reading (Boston et al., 2011; Demberg & Keller, 2008).

Surprisal theory made several steps to connect its syntactic predictions to other cases of retrieval and cognition in general (Levy, 2008; Smith & Levy, 2013) but I think it is fair to say that the strength of the theory lies in capturing expectation effects driven by the syntactic context. Consequently, it is not restricted by properties of retrieval outside of language when quantitatively fitting reading data and it is usually not used to capture processing difficulties due to the recall of dependents. This has changed recently in lossy-context surprisal (Futrell &

Levy, 2017; Futrell, Gibson, & Levy, 2020), which shows that an account building on surprisal can provide one framework for both the recall of dependents and expectation-driven effects. This computational-level approach, in contrast to the algorithmic-level approach developed in this paper, expands suprisal theory with an extra component (noisy context) to capture memory-driven difficulties. This complements the current approach, which expands memory-driven analyses of parsing with an extra component (insights from transition-based parsing) to capture expectation-driven effects on parsing.

5.2. ACT-R models of reading

Cognitive architectures have been used in previous work to model parsing and the ACT-R cognitive architecture has been the most popular choice, see Brasoveanu and Dotlačil (2018, 2020); Dubey et al. (2008); Jones (2019); Lewis and Vasishth (2005); Reitter et al. (2011); Vogelzang et al. (2017),18 probably followed by SOAR (Hale, 2014; Lewis, 1993) and CAPS (Just, Carpenter, & Varma, 1999; Varma, 2016).

The approaches in ACT-R can be divided into two groups depending on how they encode syntactic knowledge. Either they assume that syntactic knowledge is stored in the declarative memory of the agent (Reitter et al., 2011, this paper) or that syntactic knowledge is present in the procedural knowledge (Brasoveanu & Dotlačil, 2018; Brasoveanu & Dotlačil, 2020; Dubey et al., 2008; Lewis & Vasishth, 2005; Jones, 2019; Vogelzang et al., 2017). The difference drives the assumptions about how behavioral measures are captured. Since the procedural system does not operate with activations, the procedural approaches would need to consider other mechanisms. In fact, it falls out from these approaches that reading times should correlate with the number of rules/parsing steps assumed (see also Kaplan, 1972), since procedural knowledge applies serially in ACT-R, so a large sequence of rules should form a bottleneck.

While there is some evidence that the number of parse steps is correlated with brain activation (Brennan & Pylkkänen, 2017; Hale, Dyer, Kuncoro, & Brennan, 2018), I am not aware of strong evidence showing that the number of parse steps is (linearly) related to reading time. It is likely for this reason that the procedural systems ignore this straightforward prediction and focus on other predictions present in their systems.

Hale (2014) investigates to what extent reading times can be predicted by the likelihood that parsing steps should be compiled into a single rule (through production compilation). That work is directly compatible with the current proposal, in fact, it can be seen as an aspect that complements the current research. While the cue-based model of parsing investigates learning of parsing in declarative memory (through activation), production compilation represents learning in the procedural memory and if correct, it could explain why only a single retrieval per word could often be assumed (see discussion in Section 4.1).

Lewis and Vasishth (2005), among others, study how the activation of partially built structures stored in declarative memory affects retrieval times of those structures. The prediction forms the core of the cue-based retrieval. It is used in Lewis and Vasishth (2005) and the following work to study the processing of dependencies. Since the current work also assumes that dependents are recalled from declarative memory, it shares this particular prediction with

Lewis and Vasishth (2005) (see also Section 4.1, in which the recall of dependents and the recall of parsing steps are combined and tested in a single model). However, the current model goes significantly beyond Lewis and Vasishth (2005) by assuming that syntactic knowledge is also stored in declarative memory and as such, recalling parsing steps is susceptible to the same principles as the retrieval of partially built phrases (dependents).

Lewis and Vasishth (2005) provide several conceptual arguments for storing syntactic knowledge in the procedural system. However, none of these arguments are evidence against the current approach, as far as I can see. First, they point out that there is experimental evidence, showing that syntactic knowledge should be kept separate from lexical knowledge. This is compatible with the current approach since syntactic and lexical knowledge are kept separate (as two independent types of chunks in declarative memory). Second, they point out that the lexicon and the grammar map into different parts of brain activations and the latter, unlike the former, activates brain areas that have been independently established in ACT-R as regions of procedural systems (Anderson, 2007). Again, this is compatible with the current approach. While the syntactic knowledge is stored in declarative memory, deploying it requires the application of procedural knowledge, for example, the procedural systems shown in Fig. 4 and Fig. 12.

Reitter et al. (2011) provide evidence that the syntactic knowledge should be part of the declarative system. One advantage is that we can straightforwardly use the same model for comprehension and production. Second, the model can account for syntactic priming effects in production. The model in this paper is compatible with the positive results of Reitter et al. (2011).

There is another dimension as to how ACT-R parsers differ from each other. Almost all existing ACT-R parsers are constructed by hand. The parser in this paper and Reitter et al. (2011) are the only parsers, as far as I know, that are data-driven. There is a clear and significant advantage to the data-driven approach. From the modeling perspective, it makes it impossible for modelers to sneak in a good fit by tweaking hand-coded parsing steps. Second, it allows one to investigate the model on a plethora of various data. Third, it provides a general link between the model and the data: we do not need to discuss the model case by case, since it is fully and explicitly described by the algorithm of the transition-based parser implemented in ACT-R, see Section 3 and Section 4. Finally, building a data-driven parser is the first necessary step in understanding the learnability of syntactic knowledge. It is impossible to even start addressing the question of how parsing is acquired if it is not data-driven.

It should be clear that the model in this paper is closest to Reitter et al. (2011). However, there are also differences between the two approaches. First, Reitter et al. (2011) use Combinatory Categorial Grammar (Steedman, 2001) and grammar-based parsing, while this paper uses context-free grammar with gaps and transition-based parsing. This is useful since it allows us to study how parsing interacts with dependency resolutions. Second, Reitter et al. (2011) build the data-driven parsing for (a few) ditransitive sentences, that is, they do not strive to generalize their approach beyond those sentences to arbitrary structures.19 Third, Reitter et al. (2011) focus on production, whereas this paper studies comprehension. Finally, Reitter et al. (2011) develop a model to generate known qualitative effects in priming, while

this paper, through the application of ACT-R models in a Bayesian framework, shows that the approach can model quantitative data patterns.

5.3. Transition-based parsers

This section briefly compares the cue-based model to other transition-based parsers. It justifies the choice of this type of parser and argues why the accuracy of the parser is sufficient (at this point).

Transition-based parsers are a class of parsers that played an important role in computational linguistics, especially for dependency grammars (see Kübler, Mcdonald, & Nivre, 2009; Nivre et al., 2007; Zhang & Clark, 2008). One advantage of transition-based parsers over graph-based parsing and grammar-based parsing is that it is fast (under standard conditions, it has linear time complexity), it is incremental, and it allows for rich feature representations (see McDonald & Nivre, 2011; Nivre, 2004). Transition-based parsers have also been applied to phrase-structure parsing at least since Kalt (2004) and Sagae and Lavie (2005). The recent neural transition-based parsers for phrase-structure building have the F1 value around 95% on the PTB section 23 (Kitaev & Klein, 2018; Liu & Zhang, 2017). Most parsers ignore gap postulation and resolution, in contrast to ours, but there are transition-based parsers that do include gaps (Coavoux & Crabbé, 2017; Coavoux & Crabbé, 2017). Transition-based parsers have also been used in computational psycholinguistics to model EEG data (Recurrent neural network grammars; Dyer, Kuncoro, Ballesteros, & Smith, 2016; Hale et al., 2018) and reading data (Boston, Hale, Kliegl, Patil, & Vasishth, 2008; Rasmussen & Schuler, 2018).20

While the high accuracy of the state-of-the-art transition-based parsing is encouraging, as it suggests that this line of parsing can eventually be used to a much more accurate parser than the one used in this paper, it should also raise worries. Why does the parser in this paper have a much lower accuracy compared to the state of the art?

There are several reasons. First, it has been found that one of the disadvantages of transition-based parsers when compared to another class of data-driven parsers, graph-based parsers, is that they get worse with increase in sentence length and increase in dependence (error propagation, McDonald & Nivre, 2011). Traditional transition-based parsers, including the parser in this paper, explore just one path. They have to greedily select what path they will follow and stick to it until the end of the sentence. Thus, early mistakes will propagate the error throughout the whole sentence. Better transition-based parsers mitigate this type of mistake through beam search or methods to recover from errors. While the adaptation of these methods could be investigated for psycholinguistics, we are primarily not interested in the best accuracy of the parser on the complex Penn Treebank sentences, but in parsing that is human-like. Indeed, it is well known that human processor also shows error propagation in parsing, as witnessed by the fact that readers struggle to recover from garden path, the longer the wrong interpretation can be held (e.g., Frazier & Rayner, 1982). Thus, it is not a priori clear that error propagation should be avoided at all costs when we turn to psycholinguistics. For example, in the manual inspection of the parser accuracy results on PTB Section 23, it was found that coordinations were often misanalyzed by the parser. The parser

always assumed local/smallest conjunction, an assumption avoided by more sophisticated parsers. This made the parser less accurate for PTB data but more in line with human parsing since it is known that the human processor prefers local attachment for coordinations (Frazier, 1987).

Another reason why we see a low accuracy is that the parser assumes a very straightforward relation between memory instances and a parsing step. A parsing step is simply stored in declarative memory and is recalled using simple relations described in Section 2.2.21 This is in contrast to complex training methods commonly assumed in current neural parsers. Relatedly, current computational parsers assume a much richer feature system: they are enriched by vector space models representing lexical information; syntactic information is usually encapsulated in 200 or more features (see Chen & Manning, 2014 for discussion, cf. the cue-based model of parsing, which postulates around 10 features).

The decision to have a simple feature model is driven by the fact that it is important to first establish that cue-based retrieval has a measurable impact on retrieval times during parsing and can be useful in predicting reading times. For that, it is preferable to keep the model as comprehensible and simple as possible; otherwise, it would not be clear whether the results reported in Section 4 are due to the cue-based retrieval model or some confound we are not interested in but is present in complex models (e.g., meaning similarity present in word vector spaces). Compare this to the case of other models of cue-based retrieval, which also started from probably an oversimplifying position of retrieval driven by a small set of binary features, rather than postulating from the start that retrieval is driven by high-dimensional vector-based lexical models.

Finally, it is worth pointing out that even though the accuracy of the parser is not very high, the examples chosen in Section 4 show that it is sufficient to be usable in psycholinguistics, as the parser delivered correct parses for the relevant experimental sentences.

Another point of improvement would be to consider transition-based parsers that do not build the structure bottom-up. There are known issues with bottom-up parsing: it accumulates elements on the stack in right-branching structures, suffers from disconnectedness, and has problems when tied to incremental interpretation (see Crocker, 1999; Resnik, 1992). For now, the choice was driven by the fact that transition-based parsing usually is combined with bottom-up parsing. It remains to be seen whether comparable or better results can be achieved with other types of parsers, notably, left-corner parsers (cf. Lewis & Vasishth, 2005; Resnik, 1992).

6. Conclusion

This paper presented a novel psycholinguistic parser, the cue-based model of parsing. It has been shown that the theory of cue-based memory systems can be combined with transition-based parsing to produce a parser that can accurately construct phrase structures and, when combined with the cognitive architecture ACT-R, can to a large extent simulate correct reading times and regressions.

Acknowledgments

Parts of this research have been presented at University of Düsseldorf, University of Groningen, Potsdam University, Utrecht University, LINGUAE seminar in Paris, and CUNY conference 2021. I would like to thank the audiences for their comments, questions, and suggestions. I would furthermore like to thank Jelmer Borst, Adrian Brasoveanu, Emmanuel Chemla, Richard Futrell, John Hale, Laura Kallmeyer, Rick Nouwen, Cory Shain, Garrett Smith, Jennifer Spenader, Shravan Vasishth, Jan Winkowski, and the reviewers and the editor of Cognitive Science. The research presented in this paper would not be possible without the Lisa Compute Cluster supported by SURFsara (www.surfsara.nl). The research was supported by the NWO grant VC.GW.17.112.

Open Research Badges

This article has earned Open Materials badge. Materials are available at https://github.com/jakdot/parser_and_memory_additionalfiles.git.

Note

1 We ignore the number cue since it does not distinguish between the (a) and (b) cases of Fig. 1.
2 We ignore the animacy cue since it does not distinguish between the (a) and (b) cases of Fig. 2.
3 The action *postulate gap* is normally ignored in transition-based parsers, so parsers only proceed by shifting and reducing (but see Coavoux & Crabbé 2017; Crabbé 2015 as an example of transition-based parsers that do consider gap resolution). Ignoring gaps is possible if the end result is a match between hand-annotated and computer-annotated parses of pronounced terminals but it would not work if we want to move from parses to actual interpretations. Ignoring gaps and their resolutions would also make the parser less useful for psycholinguistics, which often studies the effect of gap resolution on processing.
4 Unlike, for example, Roark (2001), we keep empty categories since they will be modeled by the parser.
5 Using three chunks, rather than a single chunk, to select which action should be carried out, makes the parser less sensitive to outliers and more accurate in syntactic structure building. Adding more than three chunks does not improve the accuracy of the parser. Unfortunately, retrieving three chunks, rather than a single chunk, makes the model go against the standard ACT-R assumptions (in the architecture only a single element is retrieved). I believe that this is justifiable, given the improved accuracy and the fact that some of the stringent ACT-R restrictions, assumed for much less structured and much

less complex psychology tasks compared, are hardly tenable when modeling language (see also Boston et al., 2011).

6 Label Precision is calculated as the number of correctly constructed constituents divided by the number of all constituents proposed by the parser. Label Recall is calculated as the number of correctly constructed constituents divided by the number of all constituents present in the gold standard. F1 is the harmonic mean of the two accuracy measures. For the calculation, only non-terminal constituents are used for accuracy (i.e., trivial constituents like $\langle a, DT \rangle$ are ignored so that the accuracy measures are not artificially inflated).

7 Kitaev and Klein (2018) achieve F1 of 95.13 on PTB with a pre-trained ELMo word representations. I say more about the accuracy and comparisons between this parser and the parsers in NLP in Section 5.3.

8 The code to replicate the models can be found on https://github.com/jakdot/parser_and_memory_additionalfiles.git.

9 We focus only on the relative clause regions and we stop before word 9, which is the last word in the relative clause and shows a large slowdown, possibly due to wrap-up effects. Since nothing in the model attempts to simulate wrap-up effects, one could worry that the fit of the model to the data would be driven by factors that are orthogonal to the model if we continued beyond word 8. As a check, though, another model, which included regions 2–10, was tested. The findings for regions 3–8 were not affected.

10 This is sometimes called an end-to-end modeling in ACT-R (Anderson, 2007): we do not abstract away anything; rather, we try to model the whole process that participants have to carry out in the experiment, from retrievals to key presses.

11 One difference is that in Fig. 4, *move visual attention* was run sequentially, after *retrieve parsing steps*. However, this is a mere convenience. The visual attention had to wait for the key press and the key pressing was the bottleneck in the process, so it did not matter whether we let the visual attention move concurrently with syntax, as we do in Fig. 12, or after the syntactic processing is finished.

12 The model becomes brittle if r is estimated. In particular, low values of r lead to word skipping and word skipping makes Bayesian modeling complex. In short, we would have to also model how likely word skipping is to occur, adding an extra dependent measure in the model and we would have to separately collect reading times and regressions for those instances in which no skipping took place. This significantly increases the complexity of the model and makes it less transparent how the cognitive model connects to the data.

13 The threshold estimate of t at 3.7 might seem high. However, for any criticism of that value in the model, it should be kept in mind that the absolute value of activations of parsing steps, on which t depends, is arbitrary since S, the parameter in spreading activation, is hand-selected simply to ensure that every case of spreading activation is positive.

14 Adding more random slopes led to convergence failures.

15 I am thankful to Cory Shain for providing these data.

16 In the standard ACT-R notation, used also in Section 2, $p(c)$ is not used, instead, one writes W_c and reads it as "weight" (a free parameter to be estimated).
17 In this discussion, we ignore the base activation, which measures just how accessible a chunk is independent of context. This part of activation has no counterpart in Surprisal theory.
18 The model of Lewis and Vasishth (2005) was expanded in Engelmann et al. (2013), Engelmann (2016), and Vasishth and Engelmann (to apear).
19 As far as I can see, their model relies on PTB data just to collect rule frequencies, not to train as a full-fledged data-driven parser that could be deployed to, for example, parse a corpus.
20 While these computational psycholinguistic analyses make use of transition-based parsing, they are not closely related to this work. In contrast to the current account, the cited approaches do not reconstruct the parsers inside a cognitive architecture. Their goal is different from developing a single account of cue-based dependency resolution and syntactic processing.
21 The parser could be subsumed under a case of memory-based parsing, see Daelemans, Van Den Bosch, and Zavrel (2004). However, unlike the past cases of memory-based learning, which were inspired by memory structures to deliver the best accuracy on data-driven parsing, the current approach is inspired by memory structures to connect parsing to online behavioral measures. Such a link is not possible (or even considered) in, for instance, the theory of Daelemans et al. (2004).

References

Anderson, J. R. (1990). *The adaptive character of thought*. Hillsdale, NJ: Lawrence Erlbaum Associates.

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

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036–1060. https://doi.org/10.1037/0033-295x.111.4.1036

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

Anderson, J. R., & Reder, L. M. (1999). The fan effect: New results and new theories. *Journal of Experimental Psychology: General*, *128*(2), 186–197. https://doi.org/10.1037/0096-3445.128.2.186

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, *2*(6), 396–408. https://doi.org/10.1111/j.1467-9280.1991.tb00174.x

Arnett, N., & Wagers, M. (2017). Subject encodings and retrieval interference. *Journal of Memory and Language*, *93*, 22–54. https://doi.org/10.1016/j.jml.2016.07.005

Boston, M. F., Hale, J., Kliegl, R., Patil, U., & Vasishth, S. (2008). Parsing costs as predictors of reading difficulty: An evaluation using the potsdam sentence corpus. *JEMR*, *2*(1), 1–12. https://doi.org/10.16910/jemr.2.1.1

Boston, M. F., Hale, J. T., Vasishth, S., & Kliegl, R. (2011). Parallel processing and sentence comprehension difficulty. *Language and Cognitive Processes*, *26*(3), 301–349.

Bothell, D. (2017). Act-r 7 reference manual.

Brasoveanu, A., & Dotlačil, J. (2018). An extensible framework for mechanistic processing models: From representational linguistic theories to quantitative model comparison. In *Proceedings of the 2018 International Conference on Cognitive Modelling*.

Brasoveanu, A., & Dotlačil, J. (2019). Quantitative comparison for generative theories. In *Proceedings of the 2018 Berkeley Linguistic Society 44*.

Brasoveanu, A., & Dotlačil, J. (2020). *Computational Cognitive Modeling and Linguistic Theory*. Language, Cognition, and Mind (LCAM) Series. London: Springer. https://doi.org/10.1007/978-3-030-31846-8

Brennan, J. R., & Pylkkänen, L. (2016). Meg evidence for incremental sentence composition in the anterior temporal lobe. *Cognitive Sciences*, *41*(S6), 1515–1531. https://doi.org/10.1111/cogs.12445

Chen, D., & Manning, C. D. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 740–750). Association for Computational Linguistics.

Chen, Z., & Hale, J. T. (2021). Quantifying structural and non-structural expectations in relative clause processing. *Cognitive Sciences*, *45*(1), 45. https://doi.org/10.1111/cogs.12927

Coavoux, M., & Crabbé, B. (2017a). Incremental discontinuous phrase structure parsing with the gap transition. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics* (Vol. *1*, pp. 1259–1270), Long Papers, Valencia: Association for Computational Linguistics. https://doi.org/10.18653/v1/E17-1118

Coavoux, M., & Crabbé, B. (2017). Multilingual lexicalized constituency parsing with word-level auxiliary tasks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 331–336).

Collins, M. (1997). Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting on Association for Computational Linguistics* (pp. 16–23). Association for Computational Linguistics.

Crabbé, B. (2015). Multilingual discriminative lexicalized phrase structure parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1847–1856). Association for Computational Linguistics.

Crain, S., & Steedman, M. (1985). On not being led up the garden path: The use of context by the psychological syntax processor. In D. Dowty, L. Karttunen, & A. Zwicky (Eds.), *Natural language parsing: Psychological, computational and theoretical perspectives* (pp. 320–358). Cambridge: Cambridge University Press.

Crocker, M. W. (1999). Mechanisms for sentence processing. In S. C. Garrod & M. J. Pickering (Eds.), *Language processing* (pp. 191–232). New York: Psychology Press Hove.

Cunnings, I., & Sturt, P. (2018). Retrieval interference and semantic interpretation. *Journal of Memory and Language*, *102*, 16–27. https://doi.org/10.1016/j.jml.2018.05.001

Daelemans, W., Van Den Bosch, A., & Zavrel, J. (1999). Timbl: Tilburg memory-based learner. *Machine Learning*, *34*(1/3), 11–41. https://doi.org/10.1023/a:1007585615670

Demberg, V., & Keller, F. (2008). Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, *109*(2), 193–210. https://doi.org/10.1016/j.cognition.2008.07.008

Dillon, B., Mishler, A., Sloggett, S., & Phillips, C. (2013). Contrasting intrusion profiles for agreement and anaphora: Experimental and modeling evidence. *Journal of Memory and Language*, *69*(2), 85–103.

Dotlačil, J. (2017). Building an act-r reader for eye-tracking corpus data. *Topics in Cognitive Science*, *10*(1), 144–160. https://doi.org/10.1111/tops.12315

Dubey, A., Keller, F., & Sturt, P. (2008). A probabilistic corpus-based model of syntactic parallelism. *Cognition*, *109*(3), 326–344. https://doi.org/10.1016/j.cognition.2008.09.006

Dyer, C., Kuncoro, A., Ballesteros, M., & Smith, N. A. (2016). Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 199–209)

Engelmann, F. (2016). *Toward an integrated model of sentence processing in reading* (PhD thesis). University of Potsdam, Potsdam.

Engelmann, F., Jäger, L. A., & Vasishth, S. (2017). The effect of prominence and cue association on retrieval processes: A computational account. *Cognitive Science*, *43*, e12800.

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. https://doi.org/10.1111/tops.12026

Frazier, L. (1987). Syntactic processing: Evidence from Dutch. *Natural Language & Linguistic Theory*, *5*(4), 519–559. https://doi.org/10.1007/bf00138988

Frazier, L., & Rayner, K. (1982). Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences. *Cognitive Psychology*, *14*(2), 178–210. https://doi.org/10.1016/0010-0285(82)90008-1

Futrell, R., Gibson, E., & Levy, R. P. (2020). Lossy-context surprisal: An information-theoretic model of memory effects in sentence processing. *Cognitive Science*, *44*(3), 12814.

Futrell, R., Gibson, E., Tily, H. J., Blank, I., Vishnevetsky, A., Piantadosi, S. T., & Fedorenko, E. (2018). The natural stories corpus. In *Proceedings of LREC 2018, Eleventh International Conference on Language Resources and Evaluation* (pp. 76–82)

Futrell, R., & Levy, R. (2017). Noisy-context surprisal as a human sentence processing cost model. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics* (Vol. *1*, pp. 688–698), Long Papers (Valencia). https://doi.org/10.18653/v1/E17-1065

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis*. Boca Raton, Florida: Chapman and Hall/CRC; Taylor & Francis.

Gibson, E. (1998). Linguistic complexity: Locality of syntactic dependencies. *Cognition*, *68*, 1–76.

Gibson, E. (2000). The dependency locality theory: A distance-based theory of linguistic complexity. In Marantz, A., Miyashita, Y., & O'Neil, W. (Eds.) *Image, language, brain: Papers from the first mind articulation project symposium* (pp. 95–126). Boston: The MIT Press.

Graff, D., & Finch, R. (1994). Multilingual text resources at the linguistic data consortium. In *Proceedings of the Workshop on Human Language Technology - HLT '94*. Association for Computational Linguistics.

Grodner, D., & Gibson, E. (2005). Consequences of the serial nature of linguistic input for sentenial complexity. *Cognitive Science*, *29*(2), 261–290.

Hale, J. (2001). A probabilistic early parser as a psycholinguistic model. In *Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies 2001 - NAACL '01* (pp. 159–166). Association for Computational Linguistics.

Hale, J. (2003). The information conveyed by words in sentences. *Journal of Psycholinguistic Research*, *32*(2), 101–123. https://doi.org/10.1023/a:1022492123056

Hale, J. (2010). What a rational parser would do. *Cognitive Science*, *35*(3), 399–443. https://doi.org/10.1111/j.1551-6709.2010.01145.x

Hale, J., Dyer, C., Kuncoro, A., & Brennan, J. R. (2018). Finding syntax in human encephalography with beam search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

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, MD: Paul H. Brookes Publishing.

Jäger, L. A., Chen, Z., Li, Q., Lin, C.-J. C., & Vasishth, S. (2015). The subject-relative advantage in Chinese: Evidence for expectation-based processing. *Journal of Memory and Language*, *79–80*, 97–120.

Jäger, L. A., Engelmann, F., & Vasishth, S. (2017). Similarity-based interference in sentence comprehension: Literature review and Bayesian meta-analysis. *Journal of Memory and Language*, *94*, 316–339.

Jones, S. M. (2019). *Modelling an incremental theory of Lexical Functional Grammar* (PhD thesis). University of Oxford.

Just, M. A., Carpenter, P. A., & Varma, S. (1999). Computational modeling of high-level cognition and brain function. *Human Brain Mapping*, *8*(2–3), 128–136. https://doi.org/10.1002/(SICI)1097-0193(1999)8:2/3⟨128::AID-HBM10⟩3.0.CO;2-G

Just, M. A., Carpenter, P. A., & Woolley, J. D. (1982). Paradigms and processes in reading comprehension. *Journal of Experimental Psychology: General*, *111*(2), 228–238.

Jäger, L. A., Mertzen, D., Van Dyke, J. A., & Vasishth, S. (2020). Interference patterns in subject-verb agreement and reflexives revisited: A large-sample study. *Journal of Memory and Language*, *111*, 104063. https://doi.org/10.1016/j.jml.2019.104063

Kalt, T. (2004). Induction of greedy controllers for deterministic treebank parsers. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.

Kaplan, R. M. (1972). Augmented transition networks as psychological models of sentence comprehension. *Artificial Intelligence*, *3*, 77–100. https://doi.org/10.1016/0004-3702(72)90043-4

Kitaev, N., & Klein, D. (2018). Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Klein, D., & Manning, C. D. (2003). A* parsing: Fast exact viterbi parse selection. In *Proceedings of the Human Language Technology Conference and The North American Association for Computational Linguistics (HLT-NAACL* (pp. 119–126).

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

Kübler, S., Mcdonald, R., & Nivre, J. (2009). Dependency parsing. *Synthesis Lectures on Human Language Technologies*, *2*(1), 1–127. https://doi.org/10.2200/s00169ed1v01y200901hlt002

Kush, D., Lidz, J., & Phillips, C. (2015). Relation-sensitive retrieval: Evidence from bound variable pronouns. *Journal of Memory and Language*, *82*, 18–40. https://doi.org/10.1016/j.jml.2015.02.003

Kush, D. W. (2013). *Respecting relations*:*Memory access and antecedent retrieval in incremental sentence processing* (PhD thesis). University of Maryland, College Park.

Lago, S., Shalom, D. E., Sigman, M., Lau, E. F., & Phillips, C. (2015). Agreement attraction in Spanish comprehension. *Journal of Memory and Language*, *82*, 133–149. https://doi.org/10.1016/j.jml.2015.02.002

Lebiere, C. (1999). The dynamics of cognition: An act-r model of cognitive arithmetic. *Kognitionswissenschaft*, *8*(1), 5–19. https://doi.org/10.1007/s001970050071

Levy, R. (2008). Expectation-based syntactic comprehension. *Cognition*, *106*(3), 1126–1177. https://doi.org/10.1016/j.cognition.2007.05.006

Levy, R. (2008). A noisy-channel model of rational human sentence comprehension under uncertain input. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP '08* (pp. 1055–1065). Association for Computational Linguistics.

Levy, R., Fedorenko, E., & Gibson, E. (2013). The syntactic complexity of Russian relative clauses. *Journal of Memory and Language*, *69*(4), 461–495. https://doi.org/10.1016/j.jml.2012.10.005

Lewis, R. (1993). *An architecturally-based theory of human sentence comprehension* (PhD thesis). Carnegie Mellon University, Pittsburgh, PA.

Lewis, R. L., & Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, *29*(3), 375–419.

Lewis, R. L., Vasishth, S., & Van Dyke, J. A. (2006). Computational principles of working memory in sentence comprehension. *Trends in Cognitive Sciences*, *10*(10), 447–454. https://doi.org/10.1016/j.tics.2006.08.007

Linzen, T., & Jaeger, T. F. (2015). Uncertainty and expectation in sentence processing: Evidence from subcategorization distributions. *Cognitive Sciences*, *40*(6), 1382–1411. https://doi.org/10.1111/cogs.12274

Liu, J., & Zhang, Y. (2017). In-order transition-based constituent parsing. *TACL*, *5*, 413–424.

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, *19*(2), 313–330. https://doi.org/10.21236/ada273556

Mcdonald, R., & Nivre, J. (2011). Analyzing and integrating dependency parsers. *Computational Linguistics*, *37*(1), 197–230.

Mcelree, B. (2000). Sentence comprehension is mediated by content-addressable memory structures. *Journal of Psycholinguistic Research*, *29*(2), 111–123. https://doi.org/10.1023/a:1005184709695

Mcelree, B. (2006). Accessing recent events. *Psychology of Learning and Motivation*, *46*, 155–200. https://doi.org/10.1016/s0079-7421(06)46005-9

Mcelree, B., Foraker, S., & Dyer, L. (2003). Memory structures that subserve sentence comprehension. *Journal of Memory and Language*, *48*(1), 67–91. https://doi.org/10.1016/s0749-596x(02)00515-6

Nguyen, L., Schijndel, M. V., & Schuler, W. (2012). Accurate unbounded dependency recovery using generalized categorial grammars. In *Proceedings of COLING 2012* (pp. 2125–2140).

Nicenboim, B., & Vasishth, S. (2018). Models of retrieval in sentence comprehension: A. Computational evaluation using Bayesian hierarchical modeling. *Journal of Memory and Language*, *99*, 1–34.

Nicenboim, B., Vasishth, S., Engelmann, F., & Suckow, K. (2018). Exploratory and confirmatory analyses in sentence processing: A. Case study of number interference in German. *Cognitive Science*, *42*, 1075–1100.

Nivre, J. (2004). Incrementality in deterministic dependency parsing. In: *Proceedings of the Workshop on Incremental Parsing Bringing Engineering and Cognition Together - IncrementParsing '04* (pp. 50–57). Association for Computational Linguistics.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., & Marsi, E. (2007). Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, *13*(2), 95–135. https://doi.org/10.1017/s1351324906004505

Parker, D., & Phillips, C. (2017). Reflexive attraction in comprehension is selective. *Journal of Memory and Language*, *94*, 272–290. https://doi.org/10.1016/j.jml.2017.01.002

Patil, U., Vasishth, S., & Lewis, R. L. (2016). Retrieval interference in syntactic processing: The case of reflexive binding in english. *Frontiers in Physiology*, *7*, 329. https://doi.org/10.3389/fpsyg.2016.00329

Pearlmutter, N. J., Garnsey, S. M., & Bock, K. (1999). Agreement processes in sentence comprehension. *Journal of Memory and Language*, *41*(3), 427–456. https://doi.org/10.1006/jmla.1999.2653

Rasmussen, N. E., & Schuler, W. (2017). Left-corner parsing with distributed associative memory produces surprisal and locality effects. *Cognitive Sciences*, *42*(S4), 1009–1042. https://doi.org/10.1111/cogs.12511

Reichle, E. D., Rayner, K., & Pollatsek, A. (2003). The e-z reader model of eye-movement control in reading: Comparisons to other models. *Behavioral and Brain Sciences*, *26*(4), 445–476. https://doi.org/10.1017/s0140525x03000104

Reichle, E. D., Warren, T., & Mcconnell, K. (2009). Using e-z reader to model the effects of higher level language processing on eye movements during reading. *Psychonomic Bulletin & Review*, *16*(1), 1–21. https://doi.org/10.3758/pbr.16.1.1

Reitter, D., Keller, F., & Moore, J. D. (2011). A computational cognitive model of syntactic priming. *Cognitive Science*, *35*(4), 587–637. https://doi.org/10.1111/j.1551-6709.2010.01165.x

Resnik, P. (1992). Left-corner parsing and psychological plausibility. In *Proceedings of the 14th Conference on Computational linguistics*. Association for Computational Linguistics.

Roark, B. (2001). Probabilistic top-down parsing and language modeling. *Computational Linguistics*, *27*(2), 249–276. https://doi.org/10.1162/089120101750300526

Sagae, K., & Lavie, A. (2005). A classifier-based parser with linear run-time complexity. In *Proceedings of the Ninth International Workshop on Parsing Technology - Parsing '05* (pp. 125–132). Association for Computational Linguistics.

Salvucci, D. D. (2001). An integrated model of eye movements and visual encoding. *Cognitive Systems Research*, *1*(4), 201–220. https://doi.org/10.1016/s1389-0417(00)00015-2

Schad, D. J., Betancourt, M., & Vasishth, S. (2021). Toward a principled Bayesian workflow in cognitive science. *Psychological Methods*, *26*(1), 103–126. https://doi.org/10.1037/met0000275

Shain, C., Blank, I. A., Schijndel, M. V., Schuler, W., & Fedorenko, E. (2020). Fmri reveals language-specific predictive coding during naturalistic sentence comprehension. *Neuropsychologia*, *138*, 107307. https://doi.org/10.1016/j.neuropsychologia.2019.107307

Shain, C., & Schuler, W. (2019). *Continuous-time deconvolutional regression for psycholinguistic modeling*. Center for Open Science. https://doi.org/10.31234/osf.io/whvk5

Shain, C., Blank, I. A., Schijndel, M. V., Schuler, W., & Fedorenko, E. (2019). fMRI reveals language-specific predictive coding during naturalistic sentence comprehension. In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)* (pp. 49–58). Cold Spring Harbor Laboratory.

Smith, G., & Vasishth, S. (2020). A principled approach to feature selection in models of sentence processing. *Cognitive Sciences*, *44*(12), 12918. https://doi.org/10.1111/cogs.12918

Smith, N. J., & Levy, R. (2013). The effect of word predictability on reading time is logarithmic. *Cognition*, *128*(3), 302–319. https://doi.org/10.1016/j.cognition.2013.02.013

Staub, A. (2010). Eye movements and processing difficulty in object relative clauses. *Cognition*, *116*(1), 71–86. https://doi.org/10.1016/j.cognition.2010.04.002

Staub, A. (2011). Word recognition and syntactic attachment in reading: Evidence for a staged architecture. *Journal of Experimental Psychology: General*, *140*(3), 407–433. https://doi.org/10.1037/a0023517

Steedman, M. (2001). *The Syntactic Process*. Cambridge, MA: MIT Press.

Tucker, M. A., Idrissi, A., & Almeida, D. (2015). Representing number in the real-time processing of agreement: Self-paced reading evidence from Arabic. *Frontiers in Physiology*, *6*, 347. https://doi.org/10.3389/fpsyg.2015.00347

Van Dyke, J. A. (2007). Interference effects from grammatically unavailable constituents during sentence processing. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *33*(2), 407–430. https://doi.org/10.1037/0278-7393.33.2.407

Van Dyke, J. A., & Lewis, R. L. (2003). Distinguishing effects of structure and decay on attachment and repair: A cue-based parsing account of recovery from misanalyzed ambiguities. *Journal of Memory and Language*, *49*(3), 285–316. https://doi.org/10.1016/s0749-596x(03)00081-0

Van Dyke, J. A., & Mcelree, B. (2007). Corrigendum to "retrieval interference in sentence comprehension" *journal of memory and language* 55 (2006) 157–166. *Journal of Memory and Language*, *57*(1), 150. https://doi.org/10.1016/j.jml.2007.02.002

Van Schijndel, M., & Schuler, W. (2013). An analysis of frequency-and memory-based processing costs. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 95–105).

Van Schijndel, M., & Schuler, W. (2015). Hierarchic syntax improves reading time prediction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1597–1605). Association for Computational Linguistics.

Varma, S. (2014). The CAPS family of cognitive architectures. In *The Oxford handbook of cognitive science* (pp. 49). Oxford, England; New York: Oxford University Press.

Vasishth, S., & Engelmann, F. (to appear). *Sentence comprehension as a cognitive process: A computational approach*. Cambridge: Cambridge University Press.

Vasishth, S., Mertzen, D., Jäger, L. A., & Gelman, A. (2018). The statistical significance filter leads to overoptimistic expectations of replicability. *Journal of Memory and Language*, *103*, 151–175.

Vasishth, S., Nicenboim, B., Engelmann, F., & Burchert, F. (2019). Computational models of retrieval processes in sentence processing. *Trends in Cognitive Sciences*, *23*(11), 968–982.

Villata, S., Tabor, W., & Franck, J. (2018). Encoding and retrieval interference in sentence comprehension: Evidence from agreement. *Frontiers in Physiology*, *9*, 9. https://doi.org/10.3389/fpsyg.2018.00002

Vogelzang, M., Mills, A. C., Reitter, D., Rij, J. V., Hendriks, P., & Rijn, H. V. (2017). Toward cognitively constrained models of language processing: a review. *Frontiers in Communication*, *2*, 1–11. https://doi.org/10.3389/fcomm.2017.00011

Wagers, M. W., Lau, E. F., & Phillips, C. (2009). Agreement attraction in comprehension: Representations and processes. *Journal of Memory and Language*, *61*(2), 206–237. https://doi.org/10.1016/j.jml.2009.04.002

Wu, F., Kaiser, E., & Vasishth, S. (2017). Effects of early cues on the processing of Chinese relative clauses: Evidence for experience-based theories. *Cognitive Science*, *42*, 1101–1133.

Zhang, Y., & Clark, S. (2008). A tale of two parsers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP '08* (pp. 562–571). Association for Computational Linguistics.

Appendix A: How to calculate base activation from word/rule frequencies

We want to calculate B_i from frequency. d is a free parameter and can be ignored in this discussion.

$$B_i = \log\left(\sum_{k=1}^{n} t_k^{-d}\right) \qquad (d - \text{free parameter}) \tag{A.1}$$

Consider a 15-year-old speaker. How can we estimate how often a word/rule x was used in language interactions that the speaker participated in?

First, let us notice that we know the relative frequency of x. We collect that from the British National Corpus (for words) and from the Penn Treebank corpus (for rules).

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 x was used on average based on the frequency of x. A good approximation of the number of words a speaker is exposed to per year can be found in Hart and 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. Furthermore, we assume that each word is accompanied by one parsing step, so there are as many parsing steps as words (again, this is a simplification that should not harm modeling).

We now know how we get from frequency to the number of usages of x. Simplifying again, we assume that the usages, t_k above, are evenly spread during the life span.

The procedure described here was successfully used in translating frequencies to activations and ultimately reaction times in sentence production (Reitter et al., 2011), eye-tracking reading times (Dotlačil, 2018), and reaction times in lexical decision tasks (Brasoveanu & Dotlačil, 2020).

Appendix B: Symbolic predictions of the parser for Grodner and Gibson (2005)

This appendix shows a step-by-step parsing of an example item from Grodner and Gibson (2005). The incremental parsing of an object-relative clause is given in Fig. B.1. The incremental parsing of a subject-relative clause is given in Fig. B.2. The labels are slightly simplified for presentation purposes compared to the PTB and the parser's output (the PTB often labels trees with extra information, including grammatical relations for NPs and semantic specifications for PPs). Each framed window is the parsing result after one word is finished being parsed. When a frame carries more than one tree, this represents a case in which several trees are carried in the stack of trees $\mathcal{S}$. The trees are ordered from right to left based on their order on the stack (the rightmost position is the top of the stack). What actions were used per word is specified in (35).

Actions for an object-relative clause (specifying annotated arrows in Fig. B.1) : (B.1)

1.shift, reduce-binary (label: NP)

2.shift, reduce-unary (label: WHNP), postulate gap

3.reanalyze, shift

4.shift, reduce-binary (label: NP)

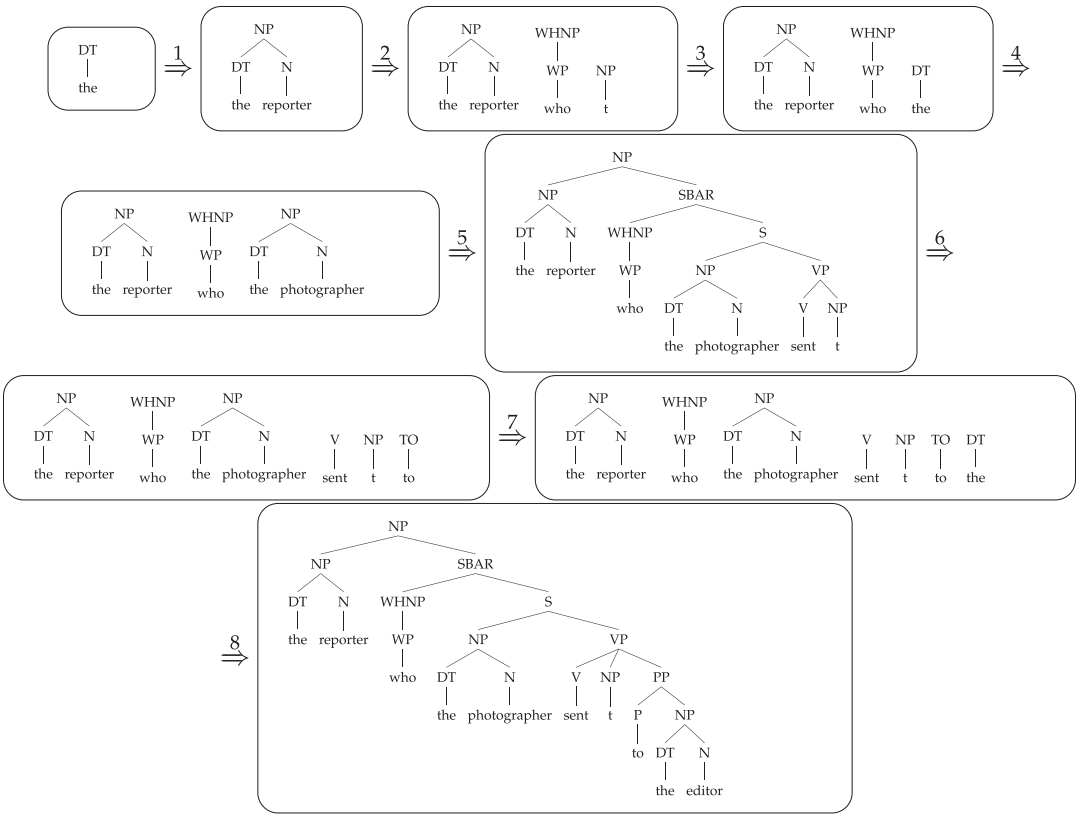


Fig. B.1. Parsing of an object-relative clause.

5.shift, postulate gap, reduce-binary (label: VP), reduce-binary (label: S), reduce-binary (label: SBAR), reduce-binary (label: NP)

6.reanalyze, shift

7.shift

8.shift, reduce-binary (label: NP), reduce-binary (label: PP), reduce-binary (label: VP), reduce-binary (label: S), reduce-binary (label: SBAR), reduce-binary (label: NP)

Actions for an object-relative clause (specifying annotated arrows in Fig. *B*.2) : (B.2)

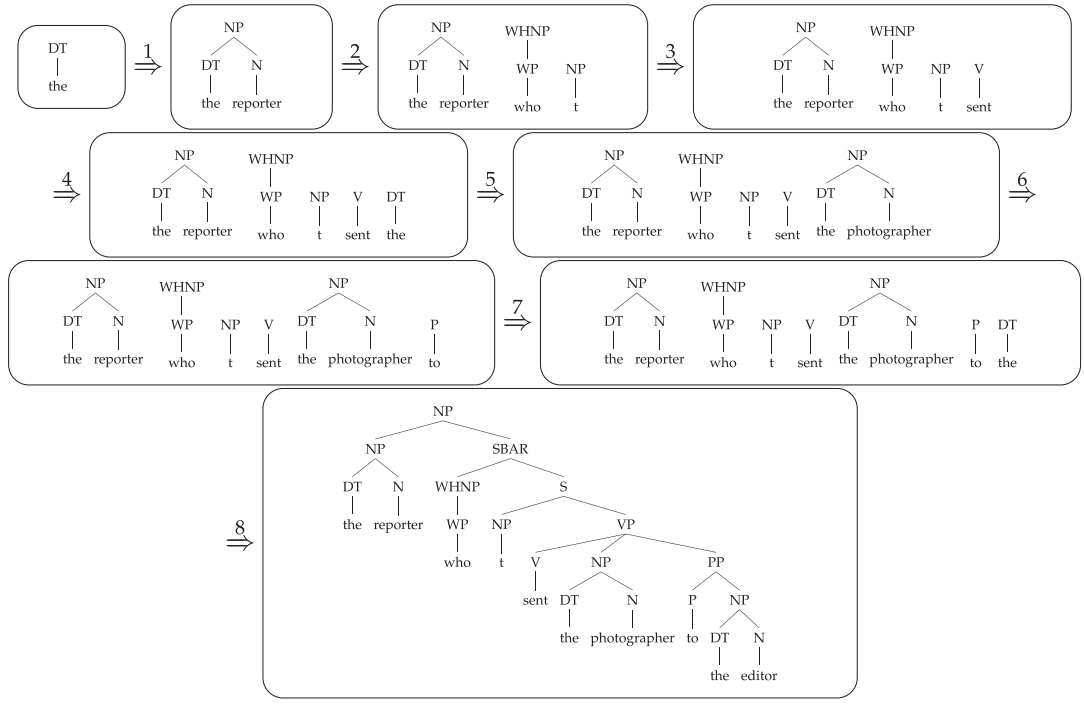


Fig. B.2. Parsing of a subject-relative clause.

1.shift, reduce-binary (label: NP)

2.shift, reduce-unary (label: WHNP), postulate gap

3.shift

4.shift

5.shift, reduce-binary (label: NP)

6.shift

7.shift

8.shift, reduce-binary (label: NP), reduce-binary
(label: PP), reduce-binary (label: VP), reduce-binary (label: S), reduce-binary
(label: SBAR), reduce-binary
(label: NP)

Appendix C: Further details on Bayesian ACT-R models for reading data of Grodner and Gibson (2005)

This section presents details of the Bayesian ACT-R models for the data from Grodner and Gibson (2005) presented in Section 4.1. Two issues are covered. We investigate prior predictive distribution of the model and its robustness, that is, whether the model can also be fitted to simulated values based on Grodner and Gibson (2005).

In prior predictive, we simulate hypothetical data based solely on the priors of the parameters of the model, as specified in Section 4.1.4. The simulations are created for the three models presented in Section 4.1: the data-driven cue-based model of parsing (Model 1), the syntactic model without Active Filler Strategy (Model 2) and the syntax-free model (Model 3). The simulations were run for 1,500 iterations. They are graphically summarized in Fig. C.1.

The prior checks for the three models are close to each other and the 95% credible intervals include mean RTs of all regions. This shows that the priors for the parameters do not a priori disadvantage one model over another when fitting to the data. The 95% credible intervals in the prior predictive checks cover mean RTs from roughly 150 ms to, in some cases, more than 2,000 ms. The upper limit might seem too benevolent and could be further restricted to match more closely the domain expertise (see, e.g., Schad, Betancourt, & Vasishth, 2019). However, it was decided not to restrict this upper limit further. This is because there are three parameters in ACT-R model that affect reading times: F, f, and W_j. It is not clear a priori which of these three parameters should be more limited in its range.

Fig. C.1 also shows that on some words, the 95% credible intervals are wider than on others. The wider intervals are observed on content words. These are regions in which every item is lexicalized differently and in which lexical frequencies can strongly differ between different items. Since the model is sensitive to frequency, it will show large variations in those regions.

Next, we check the robustness of the model (see also Schad et al., 2019). We want to see whether it can also be fit to data that are simulated from Grodner and Gibson (2005) based on standard procedures. Ideally, we should observe that the fit to such simulated data should be comparably good as the fit of the model to the actual data from Grodner and Gibson (2005). We proceed as follows: (i) we fit a linear mixed model to the data from Grodner and Gibson (2005); the model includes intercept, word position (factor with six levels), and type of relative clause (subject vs. object) as fixed effects; it also includes subjects and items random factors; (ii) we extract all parameter estimates from the linear mixed model; (iii) we simulate new data based on these estimates; and (iv) we fit a Bayesian ACT-R model to the newly simulated data. We repeat this procedure for 10 different simulated data sets.

It turns out that the Bayesian model is quite robust in the sense that it can be fit well to the values simulated according to the just given procedure. On average, the models include simulated mean RTs in their 95% credible intervals of posterior predictive distribution in 81% of cases. That is, on average, 10 out of 12 simulated mean RTs fall in the 95% credible intervals. Four selected examples of Bayesian models and their fit to simulated data sets are given in Fig. C.2.

The posterior distributions for the five parameters of the ACT-R model after the fit to the simulated data are shown in Fig. C.3. The distributions are summarized also here:

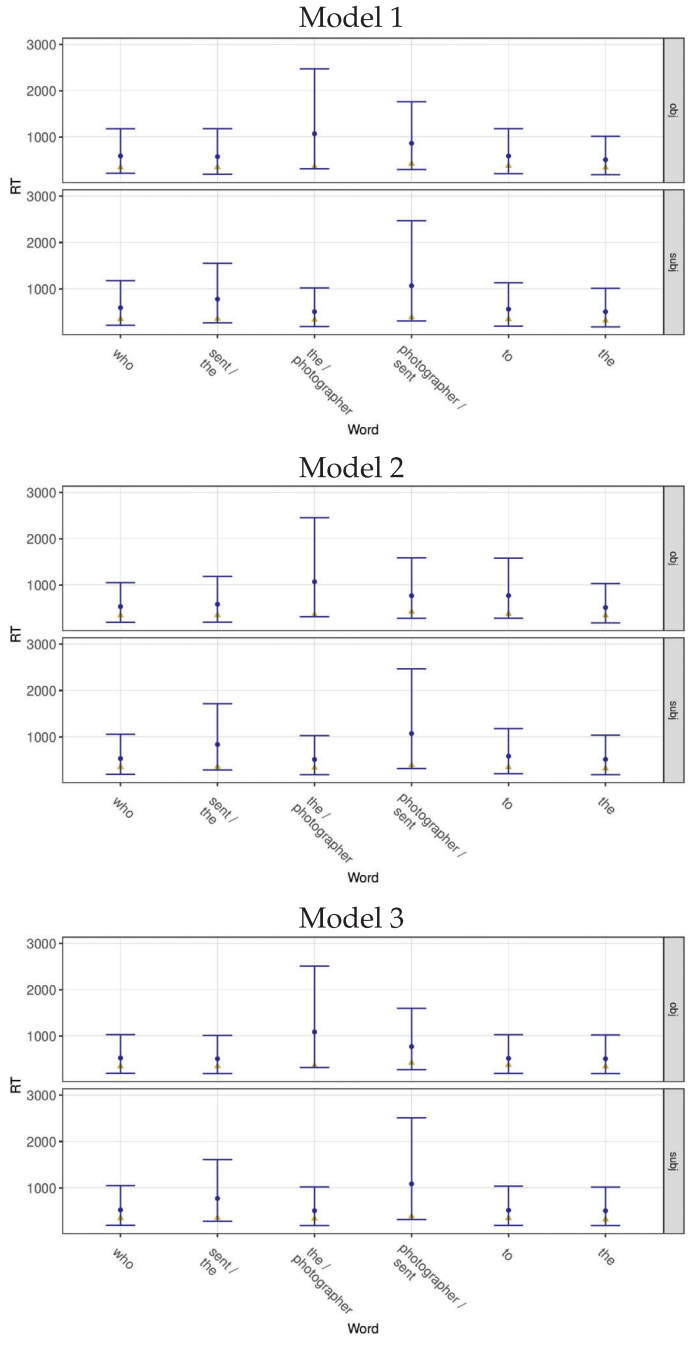


Fig. C.1. Prior predictive for the models 1–3 of Grodner and Gibson (2005). Recall that model 1 includes syntactic information, model 2 postpones trace resolution, and model 3 is syntax-free. The dots are predicted mean RTs. The bars provide the 95% credible intervals. The yellow triangles are observed mean RTs, taken from Grodner and Gibson (2005).

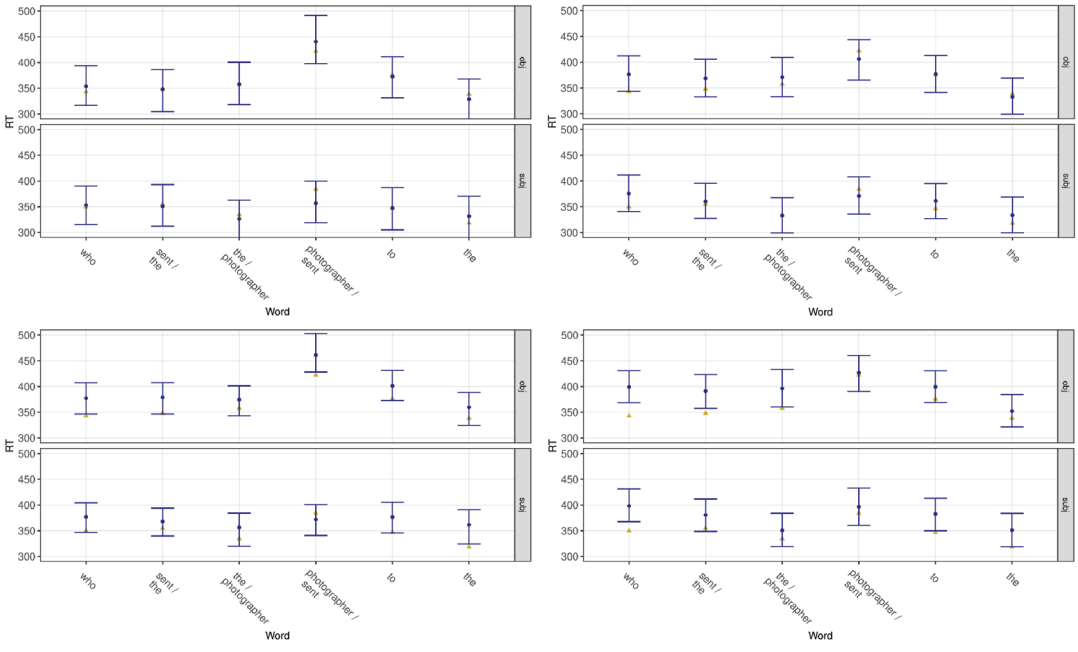


Fig. C.2. Posterior predictive for model 1 against simulated data. The data were simulated according to the procedure described in the text. The dots are predicted mean RTs. The bars provide the 95% credible intervals. The yellow triangles are mean RTs of the generated data. Four examples are selected. The top two cases represent a good fit (all or all but one simulated data fall inside the 95% credible intervals), and the bottom two cases represent a worse fit (three or more simulated data do not fall inside the 95% credible intervals).

- F—median: 0.06, sd: 0.03
- f—median: 0.12, sd: 0.1
- r—median: 0.03, sd: 0.01
- W_j—median: 33, sd: 32
- SD—median: 17, sd: 6

The posterior values of the parameters come close to the values as given in the main text based on the actual data.

In sum, we see that our Bayesian model is robust enough to generalize to new similar data generated from the estimates based on Grodner and Gibson (2005).

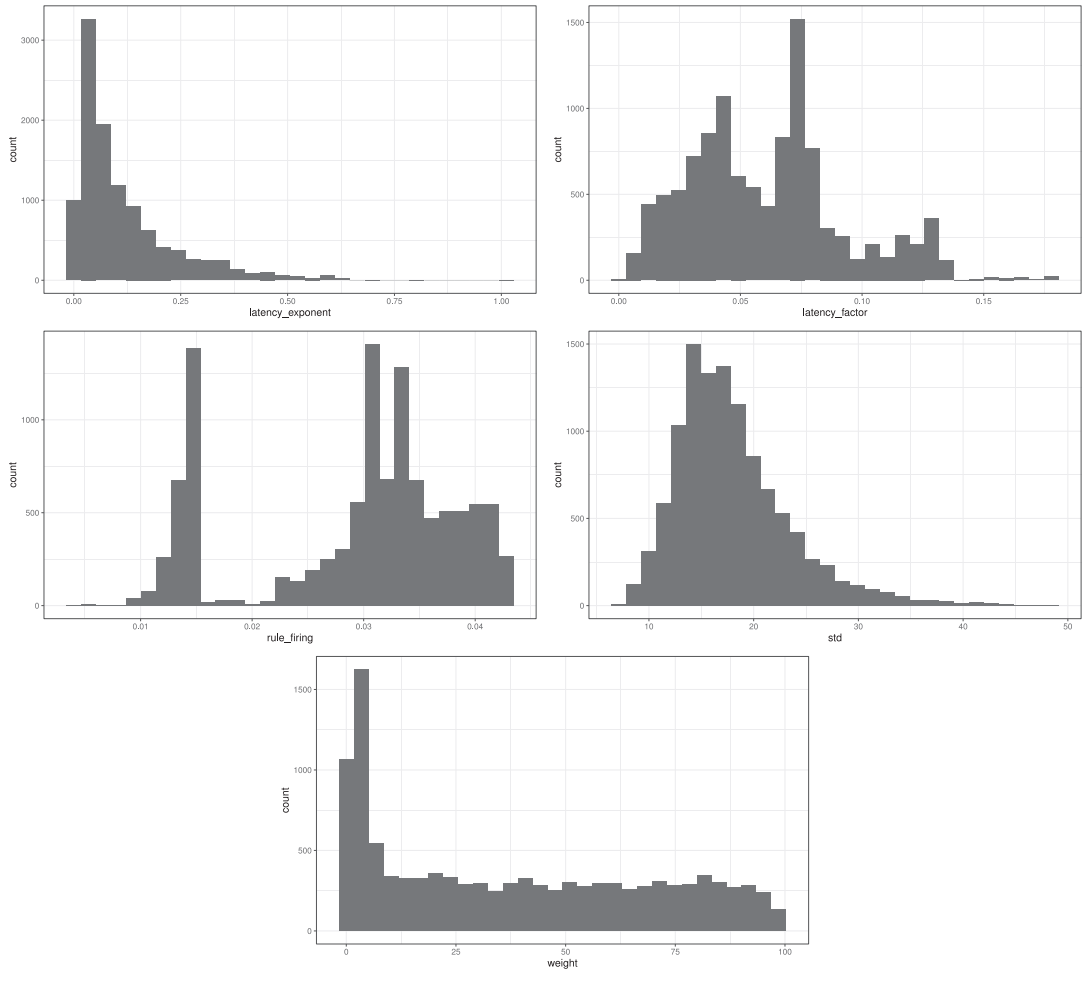


Fig. C.3. Posteriors for the five parameters estimated in the bayesian ACT-R model based on the simulated data.