Mparse: A new framework for self-organized, incremental sentence comprehension

February 13, 2021

Abstract

In the last thirty years, there have been a number of attempts to explain human sentence processing as a self-organizing process, where a parse for a whole sentence emerges purely through local, word-word interactions. Previous attempts have typically used opaque mathematical formalisms and/or failed to account for important reading time data, and so other approaches (based on rule-based parsing systems) have remained the focus of much research in sentence processing. Here, we present a new framework for self-organized sentence parsing that improves over previous implementations and opens the door to highly detailed empirical predictions. We test the new model, called *mparse*, on three important reading time effects using English materials: garden paths, local coherence, and the ambiguity advantage. Mparse successfully reproduces these effects and demonstrates the promise of its more tractable mathematical framing. We hope that the framework will stimulate new experimental and modeling work.

1 Introduction

Self-organization of syntactic structures—the idea that a parse of a whole sentence can arise solely through interactions between pairs of words—has been floating around in the sentence processing literature for at least 30 years, starting with Kempen and Vosse (1989). Under self-organization, the global parse is not constrained to be completely grammatical as in most other theories of sentence processing. Instead, globally coherent parses arise through local optimization: If there are two alternative ways of integrating a word into the preceding structure, one grammatical and one ungrammatical, the grammatical one is more likely to form, but the ungrammatical one is not prevented from forming. Local feedback loops reinforce structures as they form, so if the system ends up in a grammatical state, it is likely

to stick with it. But the same thing can happen if the system begins building a less than optimal state, which can lead to processing difficulty. Overall, self-organization says that human sentence processing is guided by grammatical rules without being bound by them.

This assumption contrasts with the typical assumption that people only entertain grammatical structures when comprehending sentences (e.g., surprisal theory and its extensions, Futrell, Gibson, and Levy 2020; Futrell and Levy 2017; Hale 2001; R. Levy 2008a, 2008b; and others Gibson 2006; Hale 2011). While these theories have been largely successful in explaining and predicting sentence comprehension behavior, the restriction to fully grammatical parses makes explaining certain facts difficult. For example, reading sentences with ungrammatical subject-verb number agreement is made less difficult if the sentence contains a non-subject distractor noun that does agree with the verb in number (Dillon, Mishler, Sloggett, & Phillips, 2013; Lago, Shalom, Sigman, Lau, & Phillips, 2015; Pearlmutter, Garnsey, & Bock, 1999; Wagers, Lau, & Phillips, 2009). Explaining this fact either requires erroneous retrieval of the distractor noun from memory (Jäger, Engelmann, & Vasishth, 2017; Lewis & Vasishth, 2005; Vasishth, Nicenboim, Engelmann, & Burchert, 2019) or the construction of an ungrammatical parse (Smith, Franck, & Tabor, 2018, 2021). A second finding that is difficult to explain using strictly grammatical parsing is local coherence effects, which seem to require that people entertain ungrammatical structures during sentence comprehension (Konieczny, 2005; Konieczny, Müller, Hachmann, Schwarzkopf, & Wolfer, 2009; Paape & Vasishth, 2015; Tabor, Galantucci, & Richardson, 2004). Self-organization-based models naturally allow for ungrammatical states because of the purely local nature of word-word interactions. Self-organization-based models have also had success in fitting other sentence processing effects, including length or "digging-in" effects in garden paths (Tabor & Hutchins, 2004), number agreement with pseudopartitive subjects (Smith et al., 2018), encoding interference effects in subject-verb number agreement (Smith et al., 2021), and certain effects of aphasia on sentence processing (Kempen & Vosse, 1989; Vosse & Kempen, 2000, 2009).

Despite these successes, previous implementations of self-organization for sentence processing have had a number of shortcomings that have prevented broad-coverage testing of the theory. First, there many (usually hand-tuned) free parameters (Kempen & Vosse, 1989; Smith et al., 2018; Smith & Tabor, 2018; Tabor & Hutchins, 2004). This is an issue because we do not know the full range of effects that the models predict and whether they constrain possible outcomes at all. Models that can predict any effect are not informative about the mechanisms they purport to explain (Roberts & Pashler, 2000). Second, papers describing these models often either do not report reading time predictions (Smith et al., 2018), or the models make incorrect predictions (Kempen & Vosse, 1989). Word-by-word

reading times from self-paced reading and eye-tracking are two of the main sources of data on how humans comprehend sentences, so a theory of human sentence processing should be able to explain how particular reading patterns come about. Finally, previous models are often unable to handle more than one or two sentence types with a single set of parameters (Smith et al., 2018; Smith & Tabor, 2018; Tabor & Hutchins, 2004). This leaves self-organization-based theories open to the criticism that they will not "scale up" to cover other well-known processing effects (e.g., Bicknell & Levy, 2009), let alone be able to make reading time predictions for arbitrary sentences. These shortcomings, in combination, call into question the viability of self-organization as a theory of human sentence comprehension. The self-organizing model presented here, called mparse[1], is designed to implement word-by-word self-organization of syntactic structure while addressing these concerns. The mathematical formalism of mparse will allow us to make very detailed reading time predictions and has few free parameters that we can systematically explore to discover the full range of predictions that the model can make. We hope that the work reported here will pave a way for future, large-scale, quantitative comparisons with other approaches, like surprisal (Hale, 2001; R. Levy, 2008a) or cue-based retrieval (Lewis & Vasishth, 2005).

The paper is structured as follows: First, the components of the new model are presented along with a description of the moment-by-moment processing dynamics. We then test mparse on three important sentence processing effects (in English) to illustrate how the model works: two types of garden paths (NP/S and NP/Z; Sturt, Pickering, & Crocker, 1999), local coherence effects (Tabor et al., 2004), and the ambiguity advantage (Traxler, Pickering, & Clifton, 1998; van Gompel, Pickering, Pearson, & Liversedge, 2005; van Gompel, Pickering, & Traxler, 2000, 2001). We conclude with a discussion of the results, the limitations of the model, and possibilities for future work.

2 Incremental parsing in mparse

Implementing self-organization as a theory of human parsing requires making a number of choices. Here, we lay out the choices we made about the nature of grammatical rules, how the model applies the rules, and how word by word reading times are derived in mparse while explaining how mparse works step by step. We present each piece in turn.

[1] van Kampen (2007) suggested "M-equation" as a less grandious-sounding alternative name for the so-called master equation that governs the dynamics in the present model (see Appendix A for details). The name "mparse" is derived from this suggestion, even though the term "M-equation" has not caught on outside of van Kampen's book.

2.1 Choices I: Grammar and parser states

Mparse represents grammatical knowledge using a dependency grammar consisting of typed, binary dependency links between specific head words and specific dependent words (de Marneffe & Nivre, 2019; Gaifman, 1965; Hays, 1964). (It would also be possible to implement mparse using other grammar formalisms like a constituency-based context-free grammar.) The grammar used for the simulations below is given in Table 1. The dependency links in the grammar are assumed to be a subset of the dependencies that a competent user of a language knows, including lexical and structural ambiguities. A sample dependency parse of a sentence is given in Fig. 1. Each link has a link harmony, a continuous measure of its well-formedness (higher harmony means more well-formed; Smolensky, 1986), and each (partial) parse, a combination of links, has a harmony that is a function of the harmony of the links (see below). For simplicity, all link harmonies are set to one in this paper; however, they can also be estimated from a corpus (e.g., Smith & Vasishth, 2020).

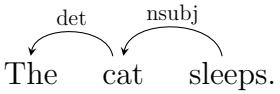

Figure 1: Dependency parse for the sentence *the cat sleeps*. Dependency types are from the Stanford Universal Dependencies (de Marneffe et al., 2014): det = determiner, nsubj = nominal subject.

Mparse uses these lexicalized binary dependency relations to create parse states that correspond to different partial or complete parses of the string so far. For a string of w words, mparse generates all partial and complete dependency parses with up to $w - 1$ dependency links using its grammar and creates a state for each one. There is also a state for the parse in which there are no dependencies between words, a no-structure state. For example, consider the following grammar with only two rules: det(cat, the) and nsubj(sleeps, cat). This grammar can generate a parse of the sentence *the cat sleeps*. Fig. 2 shows the sequence of states that are generated when reading that sentence word by word. After reading *the cat*, mparse generates two states, one with no dependency links and one with *the* attached as the determiner of *cat*. After reading *sleeps*, mparse adds two more states, one where the only dependency is that of *cat* being the subject of *sleeps* and one with the complete parse, with both *the* attached as *cat*'s determiner and *cat* as *sleeps*' subject.

Note that as more words are processed, the number of states increases rapidly; how rapidly depends on what dependencies are allowed by the grammar. Also note that the states generated by mparse need not be complete parses, and subsets of

Table 1: The grammar used in the simulations below. Direction refers to the side of the head that the dependent should be on. Words in brackets were not simulated. The harmony of each head-dependent link was set to 1.0 for all simulations. Note that *visited* can only take a direct object when it is not followed by a comma.

Sentence type	Dependency	Head	Dependent	Direction
Garden path	obj	*saw*	*doctor*	right
	ccomp	*saw*	*had*	right
	nsubj	*had*	*doctor*	left
	mark	*had*	*that*	left
	obj	*visited*	*doctor*	left
	nsubj	*had*	*doctor*	right
	advcl	*had*	*visited[,]*	right
Local coherence	obj	*smiled [at]*	*player*	right
	relcl	*player*	*tossed*	right
	relcl	*player*	*thrown*	right
	nsubj	*tossed*	*player*	left
Ambiguity advantage	nmod	*driver*	*[of] car*	right
	nmod	*car*	*[of] driver*	right
	nmod	*[of] driver*	*[with] mustache*	right
	nmod	*son*	*[of] driver*	right
	nmod	*son*	*[with] mustache*	right

nsubj = nominal subject; obj = direct object; ccomp = clausal complement; mark = relative clause marker; advcl = adverbial clausal modifier; relcl = relative clause modifier; nmod = nominal modifier

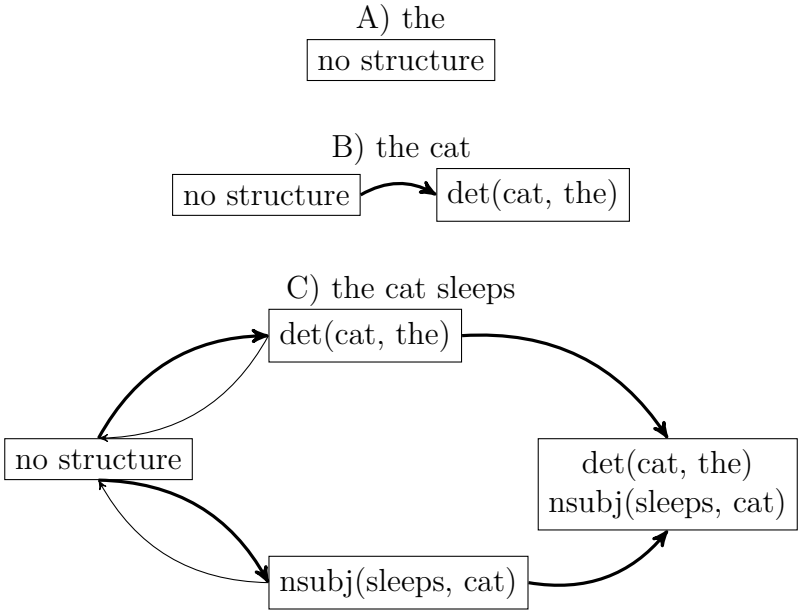

Figure 2: Network representation of the states for *the cat sleeps*. The arrows indicate possible transitions, with thicker arrows indicating a higher transition rate (higher probability of transitioning per unit time) than thinner arrows.

the links in the most complete parse (i.e., partial parses) are themselves separate states of the system.A number of checks are in place to make sure that each state is a valid dependency structure. First, core arguments of head words, e.g., subjects of verbs, determiners of nouns, etc., can only be used once per head word: A verb can only have one subject, and a noun can only have one determiner. Second, each dependent word has at most one head word governing it. Third, there can be no cycles in the dependency structure, i.e., a word cannot be its own dependent or a dependent of one of its dependents.

Finally, for a head-dependent link to be a part of a parse, it must appear in the grammar. This differs from the assumptions of some other self-organizing models like Smith et al. (2018, 2021); Smith and Tabor (2018), which allow dependency relations between any two words in a sentence. This divergence from previous self-organizing models has a number of motivations: very ungrammatical attachments (like attaching the determiner *the* as the subject of a verb) likely play little to no role in human sentence processing. Competent language users presumably know that they can ignore such terrible configurations in most circumstances. Also, it is simply more practical from an implementation point of view to only use dependency triples specified in a grammar. One does not need to make decisions about the harmony values of grammatically impossible configurations, and the processing

dynamics are easier to understand.

The spirit of self-organization is preserved in mparse because the partial and complete parses are built without considering whether they are globally coherent. Each individual dependency link must be in mparse's grammar, but the overall configuration of links need not be mutually consistent. All possible states that meet the constraints are constructed and considered without any input from an "overseer." This is the key to explaining local coherence effects, but it also plays a role in the garden path and ambiguity advantage examples below.

2.2 Choices II: State harmonies and transition rates

Even though mparse considers less than perfect structures, that does not mean that it treats all states the same. In addition to the link harmonies, whole states, which consist of zero or more links, are assigned harmony values. The harmony of a state, as implemented here, is calculated based on three pieces of information. First, a state has higher harmony if the dependency links in it have higher harmonies themselves. Since all of the links have a harmony of 1.0, states that contain more links have a higher harmony than those with fewer links. Second, in dependency grammar, a complete parse of sequence of w words has to contain $w-1$ dependency links. Thus, states with more or less than $w-1$ links are penalized by decreasing their harmony. Third, the word order preferences given by the grammar affect the harmony of a state, as well. If a word attaches as a dependent of some head word, but the order of the head and the dependent does not accord with the head's preference, the harmony is penalized. Finally, dependency length also affects a state's harmony. The more words between a head and its dependent, the more the state's harmony is penalized.

Thus, mparse constructs a discrete state space where each state is a combination of dependency links between words in the sentence so far. At each word, new states are added because the new word introduces new ways of interacting with the words that have already been input. Each state is assigned a harmony that reflects how well-formed that state is.

We now describe how mparse explores its states and makes reading time predictions. While processing each word in a sentence, mparse jumps randomly between states. This jumping around between states constitutes "parsing" in mparse. One can think of the model as considering different analyses of the sentence one by one in a random sequence. It continues this until it reaches a state whose dependency parse is as complete as possible. "As complete as possible" means that a parse contains $w-1$ dependency links for a string of w words.[2]

[2]Sometimes reading the w-th word leads to a situation where there are no structures possible with $w-1$ links. In that case, mparse processes the word until it has built a structure with the

In general, there can be more than one structure with $w-1$ links, some more grammatical, some less grammatical. But once it has found *some* state with $w-1$ links, it reads in the next word.

Mparse jumps stochastically between parse states, but not all jumps are allowed. It can only move to a new state if the new state differs from the current state only by a single dependency link. This reflects the assumption of self-organization where parses emerge through local word-word interactions. Adding word-word dependencies one at a time embodies this locality. Also, not all jumps are equally likely: jumps to more well-formed states are more probable than jumps to less well-formed states (note the different arrow thicknesses in Fig. 2). More specifically, the transition rate (probability of jumping per unit time) is higher when jumping from a low-harmony state to a higher harmony state than in the other direction. A noise parameter controls how preferable well-formed states are over ill-formed ones, with low noise resulting in a strong preference for jumping to well-formed states (compared to ill-formed states) and high noise resulting in jumps to well- and ill-formed states occurring with more equal probability.

Mathematically, mparse uses a random walk to explore possible parses of a string of words. The random walk can be described using the master equation (van Kampen, 2007), which describes how the probabilities of a system being in different discrete states changes in continuous time. This mathematical formalism was developed in physics and chemistry (see the papers reprinted in Oppenheim, Shuler, & Weiss, 1977, for examples), and its properties have been well-understood for decades. The master equation is also employed in an influential model of eye movement control, SWIFT (Engbert, Longtin, & Kliegl, 2002; Engbert, Nuthmann, Richter, & Kliegl, 2005). Here, we take advantage of this well-established mathematical apparatus for sentence parsing. Appendix A provides a more detailed description of how the master equation works.

2.3 Choices III: Starting and stopping

With a set of states and a means of moving between them, we now must specify how mparse starts and stops processing a word, which will allow us to make processing time predictions for each word in a sentence. At the first word in a sentence, there is no dependency structure that mparse can build. Thus, it just "builds" the no-structure state and then inputs the next word. After that, it adds new states based on how the new word can interact with the previous one, calculates the harmonies of the new states and the transition rates between them, and begins jumping around. It stops processing a word once it reaches an *absorbing state*, a state with $w-1$ dependency links for a string of w words. The state on the far

most dependency links possible given the words so far.

right of Fig. 2 (C) is an example; it has two dependency links for a string of three words. The amount of time it takes to reach any absorbing state is taken to be mparse's prediction for the reading time of that word. After reaching an absorbing state, the next word is input, and the process repeats again until there are no more words to input.

The mathematical formalism of mparse makes it simple to calculate mean processing time at each word in a sentence using the matrix of transition rates between states (see Appendix A). The experiments below take advantage of this and compare the patterns of mean reading time predictions in mparse to the qualitative patterns of results (ordering of conditions) from existing sentence comprehension experiments.

3 Experiments

In this section, we test mparse on three important classes of processing time effects in sentence comprehension: the contrast between two types of garden paths, local coherence effects, and the ambiguity advantage. In order to understand the full set of predictions mparse can make and to determine whether there are processing effects which it *cannot* model, we vary its noise parameter T over a wide range. Simulations of this sort are important in evaluating how informative the parser is as a theory of sentence processing (Roberts & Pashler, 2000): If there are parameter settings that allow the model to predict any possible ordering of condition means, then it cannot be said to "predict" the pattern we actually observe in human data. Only if the model rules out some possible outcomes can it same something about how the process it models might actually work.[3]

3.1 Garden paths

Garden path effects occur when a temporary ambiguity leads to processing difficulty at disambiguation. They are an empirically well-established sentence comprehension effect (Bever, 1970; Frazier & Fodor, 1978; Kimball, 1973). Here, we focus on the reported difference in magnitude between two types of garden paths, NP/S and NP/Z (Grodner, Gibson, Argaman, & Babyonyshev, 2003; Prasad & Linzen, 2019; Sturt & Crocker, 1997; Sturt et al., 1999). In NP/S garden paths, a phrase is temporarily ambiguous between a noun phrase complement (NP) of an optionally transitive verb and the subject of a sentential complement of the verb (S):

[3]Mparse contains a second free parameter τ that sets the numerical scale on which processing times take place. This parameter does not affect the ordering of conditions or standardized effect size measures, so it is set to 1.0 for all simulations. Future work will estimate τ from reading time corpora to put mparse's predictions on the millisecond scale.

(1) NP/S ambiguity:

a. The woman saw the doctor had been drinking.
b. The woman saw that the doctor had been drinking.

When a person reads (1-a) word by word, the noun phrase *the doctor* is temporarily interpreted as the direct object of *saw*. But once the reader gets to the verb phrase *had been drinking*, it becomes clear that *the doctor* needs to be the subject of the second verb phrase (see Fig. 6 (A) and (B)). This reanalysis has been observed to cause reading time slowdowns in the second verb phrase compared to (1-b) (Grodner et al., 2003; Prasad & Linzen, 2019; Sturt et al., 1999). In (1-b), reading *that* prevents attaching *the doctor* as *saw*'s direct object, and the correct parse can be built quickly.

A similar effect occurs in (2).

(2) NP/Z ambiguity:

a. Before the woman visited the doctor had been drinking.
b. Before the woman visited, the doctor had been drinking.

When a person reads (2-a), it is possible to attach *the doctor* as the direct object of *visited*. But after reading *had been drinking*, it is clear that *the doctor* has to be the subject of the second clause instead, leaving *visited* with no ("zero") complement (see Fig. 6 (C) and (D)). This causes reading time slowdowns compared to (2-b), where the comma after *visited* makes it clear that *the doctor* cannot act as the direct object.

A common assumption is that large changes to existing structure are more costly than small changes. This assumption is expressed in various reanalysis-based parsing strategies (see Sturt et al., 1999, for discussion) and parallel models (e.g., R. Levy, 2008a). In the case of NP/S and NP/Z garden paths, this predicts that NP/S ambiguities should be more difficult to resolve than NP/Z ambiguities. Sturt et al. (1999) present one way of making this assumption concrete by arguing that reanalyses that break established dominance relations are more costly than reanalyses that preserve them. Consider Fig. 6: (A) and (C) show garden path structures for the NP/S and NP/Z sentences, respectively; (B) and (D) show the correct parses after reading the next word *had*. In both (A) and (B), the main clause verb *saw* dominates the word *doctor* through a chain of dependencies, both in the garden path state and in the correct parse. However, the dominance relation between *visited* and *doctor* in (C) has to be broken in order to build the correct parse in (D). Sturt et al. (1999) argue that it is this dominance-breaking reanalysis that causes the additional slowdown for NP/Z garden paths.

We tested mparse on these two types of garden paths. As shown below, mparse produces the expected difference in effect sizes, but the reasons are somewhat

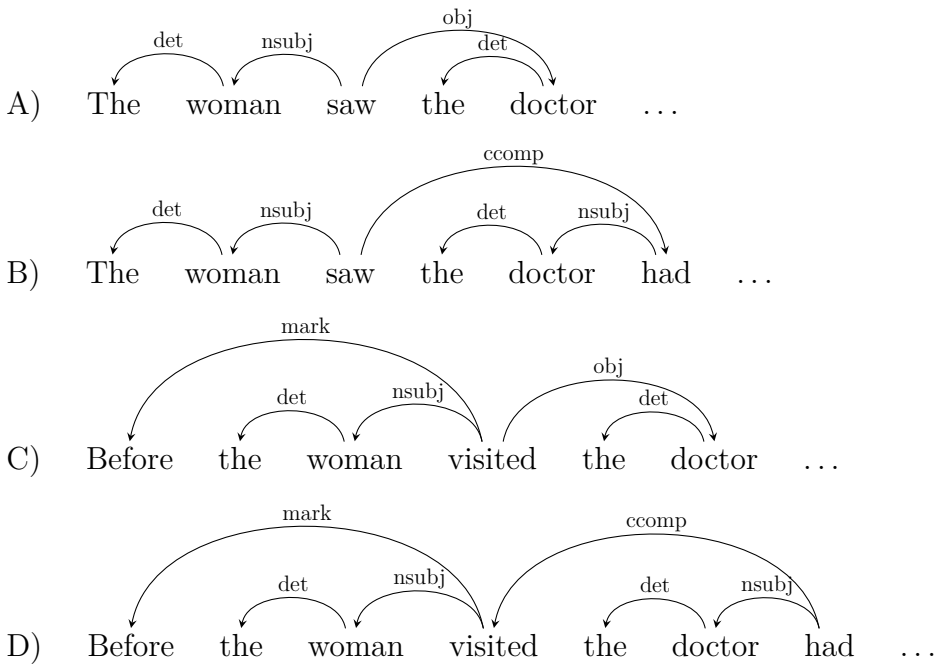

Figure 3: Dependency parses for NP/S (A and B) and NP/Z ambiguities (C and D). A and C show only temporarily viable parses where the noun *doctor* attaches as the direct object (obj) of the preceding verb. After reading the auxiliary *had*, the garden path parses must be revised in order to build the correct parses, shown in B and D. ccomp = clausal complement; mark = complementizer or subordinating conjunction.

different than those put forward by Sturt et al. (1999).

3.1.1 Method

To help in understanding how mparse processes the crucial aspects of the materials, simplified materials were used (NP/S in (3-a) and (3-b), NP/Z in (3-c) and (3-d)):

(3) a. saw doctor had (NP/S)
b. saw that doctor had (NP/S control)
c. visited doctor had (NP/Z)
d. visited, doctor had (NP/Z control)

These simplified materials retain the essential properties of the full sentences, but they require few enough mparse states that the model's processing can be easily visualized (see below). Word-by-word model output for the full sentences in (2) is provided in Appendix B. Mean reading times from mparse were calculated using the grammar shown in Table 1.

3.1.2 Results and discussion

The mean processing times at the final words in (3) are plotted in Fig. 4. For all values of the noise parameter T, the garden path conditions (labeled "ambiguous") are processed more slowly than the unambiguous control conditions. Mparse thus reproduces one of the foundational reading time effects in sentence processing research. The mean processing time curves for both garden path conditions are identical, so only the NP/Z curve is visible.

Note that the mean processing time curves for both garden path conditions and the NP/S control condition diverge to positive infinity as T approaches zero. This is due to the existence of the dead-end garden path parse states in these materials (see Fig. 5 (A) through (C)). These states have relatively low harmony, but their harmony is still higher than the no-structure state. Recall that the transition rates between states are a function of their relative harmonies and the noise level. Mparse is less likely to jump from a higher-harmony state to a lower-harmony state than the other way around, and this property is exaggerated as the noise level decreases. It becomes less and less likely that the model will jump out of a garden path state to the lower-harmony no-structure state. Thus, as the noise decreases, mparse must spend longer and longer amounts of time in the garden path state (if it jumps there), which drives the average processing times up sharply. This does not happen in the NP/Z control case, which does not contain any dead-end states (see Fig. 5 (D)) because the grammar indicates that *visited* followed by a comma does not take a direct object.

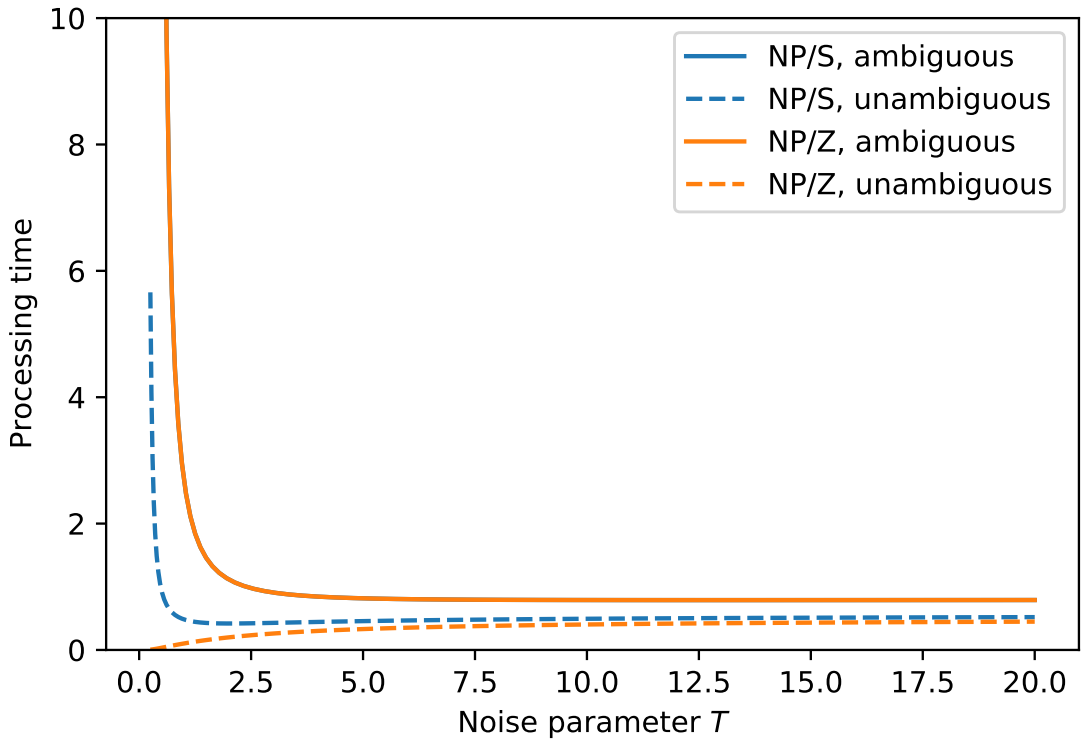

Figure 4: Mean processing times for NP/S and NP/Z garden path materials. The processing times for the ambiguous conditions in both types of sentence are identical, so only the curve for NP/Z is visible.

Obviously, people's reading times in the disambiguating region of a garden path sentence are not infinite, so we suspect that very low noise levels are not plausible, although they are included here for completeness. Very low noise implies a near inability to backtrack or reanalyze a string. We know that reanalysis is possible in most cases, so in future work, when T is fit to human data, we expect that the fitted values will not be in this very low range.

Returning to the difference between the two types of garden paths, the effect size of the NP/S effect is smaller than the NP/Z effect size for the full range of T tested. This is also related to the existence of dead-end garden path states shown in Fig. 5. For the NP/S conditions (shown in blue in Fig. 4), both the ambiguous and unambiguous conditions contain dead-end garden path states that slow average reading times. The NP/S control condition contains a number of different paths (via states that attach *that* as the relative clause marker of *had*) to the absorbing state shown on the far right of Fig. 5 (B). If, for example, mparse happened to be in the no-structure state, there are four possible paths forward, but only one leads to the dead end. Compare this to the state network for the ambiguous NP/S condition in Fig. 5 (A), which has only three ways out of the no structure state. If we further assume that each way out of the no-structure state is equally probable (which, in general, need not be the case), the probability of getting garden pathed is 0.25 in the control condition and 0.33 in the ambiguous condition. Mparse can always get garden pathed in NP/S materials, but the lower probability of it happening in the control condition makes its average processing time faster.

The situation is somewhat different for the NP/Z conditions (plotted in orange in Fig. 4. The state network for the ambiguous NP/Z sentence is identical to that of the ambiguous NP/S condition, so the explanation for slowed processing is the same. But the NP/Z control condition is different (compare Fig. 5 (B) and (D)). The NP/Z control condition has no dead-end garden path state, so it never needs to backtrack to get on a path to the absorbing state. With NP/S controls, though, mparse can still get garden pathed, which causes slower average reading times than in the NP/Z controls.

A difference in processing times in *control* conditions seems unusual. It certainly contrasts with the dominance-breaking explanation of Sturt et al. (1999). Further human experiments will be necessary to determine whether this prediction of the model can be confirmed.[4]

[4]Grodner et al. (2003) found a similar *numerical* pattern in some of their materials (see their Appendix B): The control conditions for the NP/S sentences were slower than the controls for NP/Z. However, this did not hold in Sturt et al. (1999) or Keller, Gunasekharan, Mayo, and Corley (2009), so new, higher-powered human experiments are needed to test mparse's prediction.

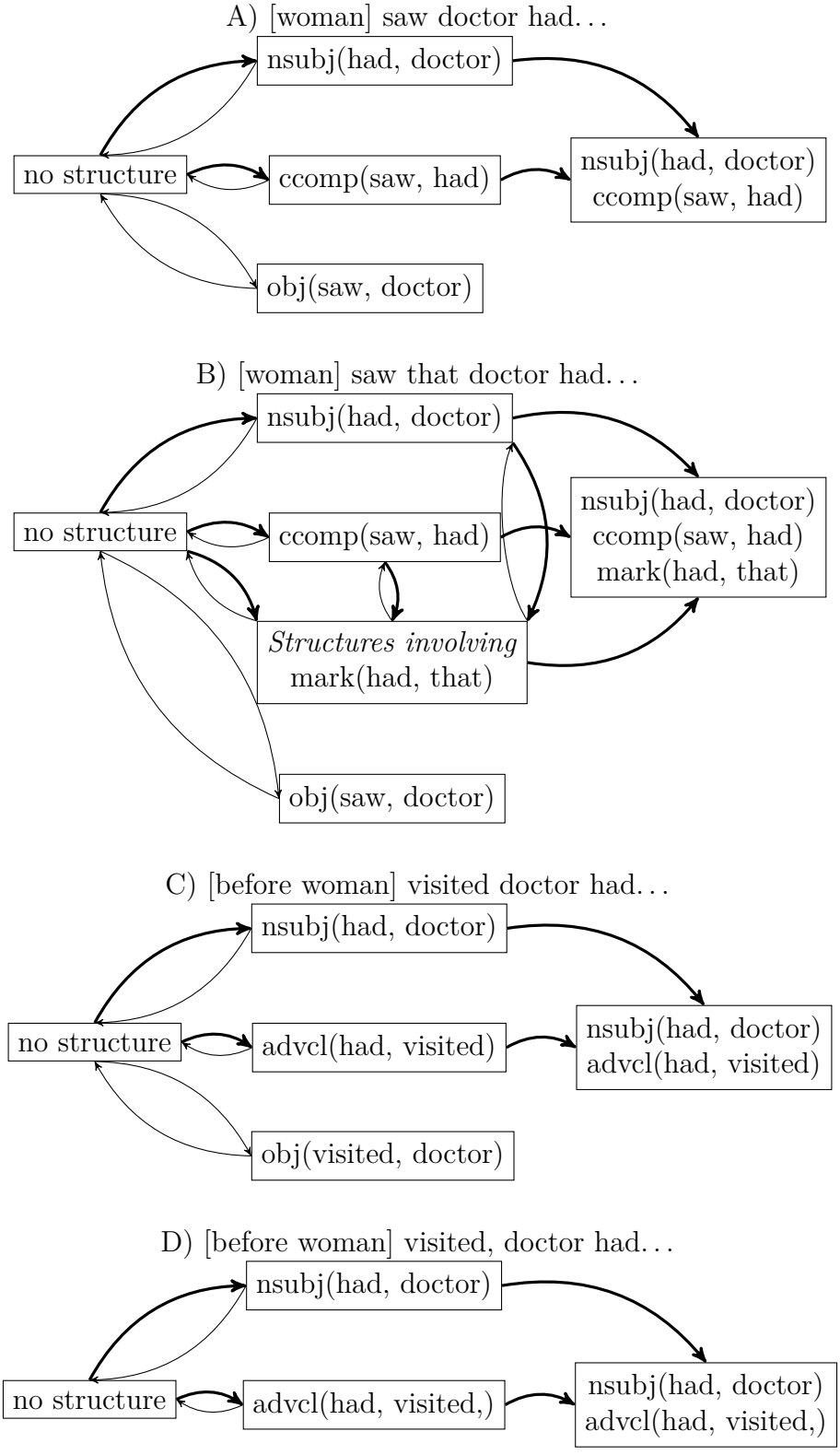

Figure 5: Network representation of the states for the reduced forms in (3). The arrows indicate possible transitions, with the thicker arrows indicating a higher transition rate than the thinner arrows. The state labeled "Structures involving mark(had, that)" in B) conflates multiple parse states where *that* attaches as the relative clause marker of *had*. Note the lack of a dead-end state in D).

3.2 Local coherence effects

Local coherence effects are one of the main motivations for considering self-organization as a theory of sentence comprehension (Tabor et al., 2004). Tabor et al. considered sentences like (4):

(4) a. The coach smiled at the player who was thrown the frisbee.
b. The coach smiled at the player thrown the frisbee.
c. The coach smiled at the player who was tossed the frisbee.
d. The coach smiled at the player tossed the frisbee.

Note that (4) (a-b) and (4) (c-d) have all have similar meanings: there is a frisbee player who caught a frisbee, and the coach smiled at that player. The (b) and (d) examples contain the reduced relative clause *. . . tossed/thrown the frisbee*, which should elicit longer reading times due to the low frequency of that structure compared to the non-reduced forms in (a) and (c). (4) (d) contains the string *. . . the player tossed the frisbee*. If this string appeared on its own, it would be a perfectly grammatical main clause with *the player* attaching as the subject of the verb *tossed*. But in the context of the preceding sentence, this analysis is not possible; the phrasal verb *smile at* cannot take a complete sentence as its complement.

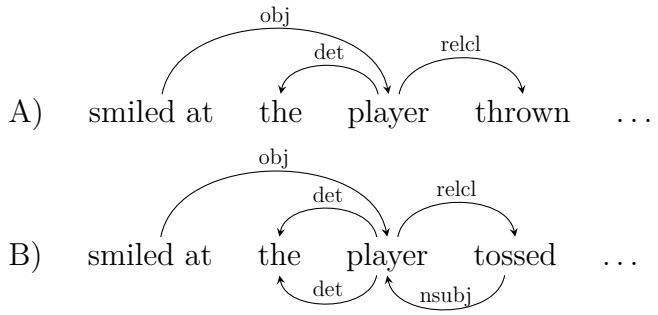

Figure 6: Dependency parses for the local coherence materials in (4). The locally coherent parse in (B) is shown below the words of the sentence. relcl = relative clause modifier.

Theories in which only globally grammatical analyses are considered would not predict a difference in processing times between (4) (b) and (d). However, self-organizing theories do because they allow merely locally acceptable structures to compete with globally grammatical ones. Tabor et al. (2004) found self-paced reading times were elevated for both (4) (b) and (d) (compared to the control sentences (a) and (c)) and that (d) was processed even more slowly than (b). This effect, which has been replicated a number of times (Christianson, Luke, Hussey, &

Wochna, 2017; Kamide & Kukona, 2018; Konieczny, 2005; Konieczny et al., 2009; R. Levy, Bicknell, Slattery, & Rayner, 2009; Müller & Konieczny, 2019; Paape & Vasishth, 2015), is taken as evidence for the ungrammatical analysis of *. . . the player tossed the frisbee* competing with the globally grammatical parse, causing slowed reading times.

Theories that posit only grammatical structures have attempted to explain local coherence through noisy-channel reinterpretation of the input string (R. Levy, 2008b; R. Levy et al., 2009) or through costly integration of bottom-up parse formation with top-down global monitoring (Bicknell & Levy, 2009; Gibson, 2006; Morgan, Keller, & Steedman, 2010). While these theories merit further discussion, we focus here on understanding how mparse explains local coherence. We do note that self-organizing theories have the benefit of parsimony in explaining local coherence effects, as the ungrammatical competitor parses arise naturally via self-organization, and additional mechanisms to not need to be added in order to explain the experimental results.

3.2.1 Method

To help in understanding how mparse processes the crucial aspects of the materials, abbreviated forms were used:

(5) a. smiled-at player who was thrown
b. smiled-at player thrown
c. smiled-at player who was tossed
d. smiled-at player tossed.

These simplified materials retain the essential properties of the full sentences, but they require few enough mparse states that the model's processing can be easily visualized (see below). Word-by-word model output for the full sentences in (4) is provided in Appendix B. Mean reading times from mparse were calculated using the grammar shown in Table 1.

3.2.2 Results and discussion

The reading times at the last words in (5) are shown in Fig. 7. The pattern of results that mparse produces is qualitatively the same as what Tabor et al. (2004) observed in self-paced reading times: The reduced forms are read slower than the non-reduced forms, and the locally coherent reduced condition is read more slowly than the non-locally coherent reduced condition.

We can understand how this effect works with the help of Fig. 8. Fig. 8 shows the states (in boxes) that are available after reading *at player tossed/thrown*. The arrows show the possible transitions between states, with thicker arrows indicating

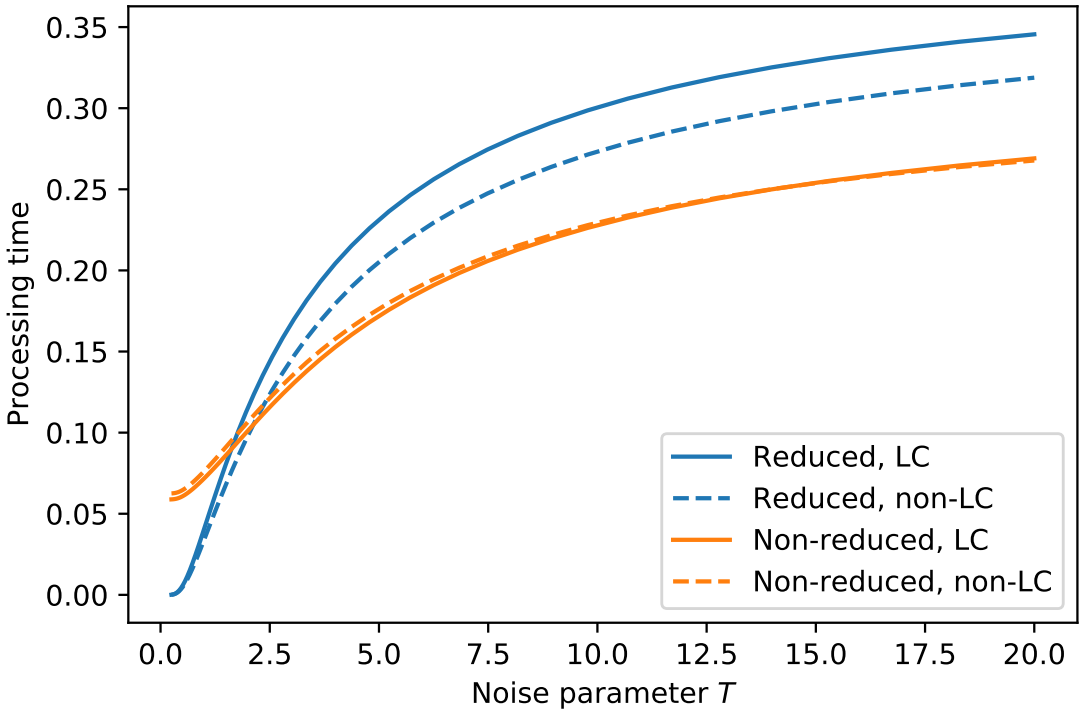

Figure 7: Processing time results from mparse for the local coherence (LC) materials in (5). Mean processing times at *tossed* or *thrown* (y-axis) are plotted in arbitrary time units as a function of the noise parameter T on the x-axis.

higher transition rates. The upper network corresponds to the *thrown* condition. If, for example, we assume that mparse was initialized in the no-structure state at the far left, it is clear that the model will have little trouble reaching the absorbing state on the far right which corresponds to the correct, full dependency parse of the words. The bottom network, which corresponds to the *tossed* condition contains a dead end. If mparse is currently in the no-structure state, it is approximately equally likely to jump to one of the states that lead to the absorbing state or to the locally coherent state. If it does jump to the latter, it will have to backtrack in order to reach the absorbing state with the correct structure. The possible need for backtracking leads to longer processing times on average compared to the *thrown* condition.

In the non-reduced conditions (e.g., *. . . who was tossed. . .*), a similar situation holds. There is a dead-end, locally coherent state in the *who was tossed* condition but not in the *who was thrown* condition. However, the dead-end state here has very low harmony because it has too few links. Thus, mparse is very unlikely to jump to it, and it only has a small impact on average reading times.

The reason that the non-reduced forms are read more quickly than the reduced forms for most noise levels is that the state networks for the non-reduced forms contain more paths to the absorbing state than the networks for the reduced forms. Having many ways of reaching the absorbing state generally speeds processing; this was also the explanation for why the NP/S control condition was faster than the NP/S garden path condition. For very low noise levels, though, the pattern flips for local coherence: The non-reduced forms are processed more slowly. This is because some of the additional paths that are available in the non-reduced forms contain lower-harmony states, specifically, those with longer-distance dependencies like nsubj(thrown, who). With low noise, the probability of jumping to lower harmony states drops rapidly, effectively removing paths to the absorbing state. The dependencies in the reduced forms are all relatively short, so the effective number of paths to the absorbing state remains approximately the same, even for low noise.

An important difference between how mparse processes garden paths and local coherence becomes apparent when contrasting Figs. 4 and 7. Processing times diverge at low noise settings for garden paths but not for local coherence materials. The reason is this: For local coherence, if mparse starts at the no-structure state in the reduced LC condition, it is approximately equally likely to take any of the three available transitions to other states. Two of those transitions lead towards the absorbing state, and those two have approximately equal harmony and are therefore approximately equally likely. For NP/S garden paths, e.g., the dead-end garden path state and the state with the nsubj(had, doc) link have approximately equal harmony, so mparse is approximately equally likely to go to one of them. But

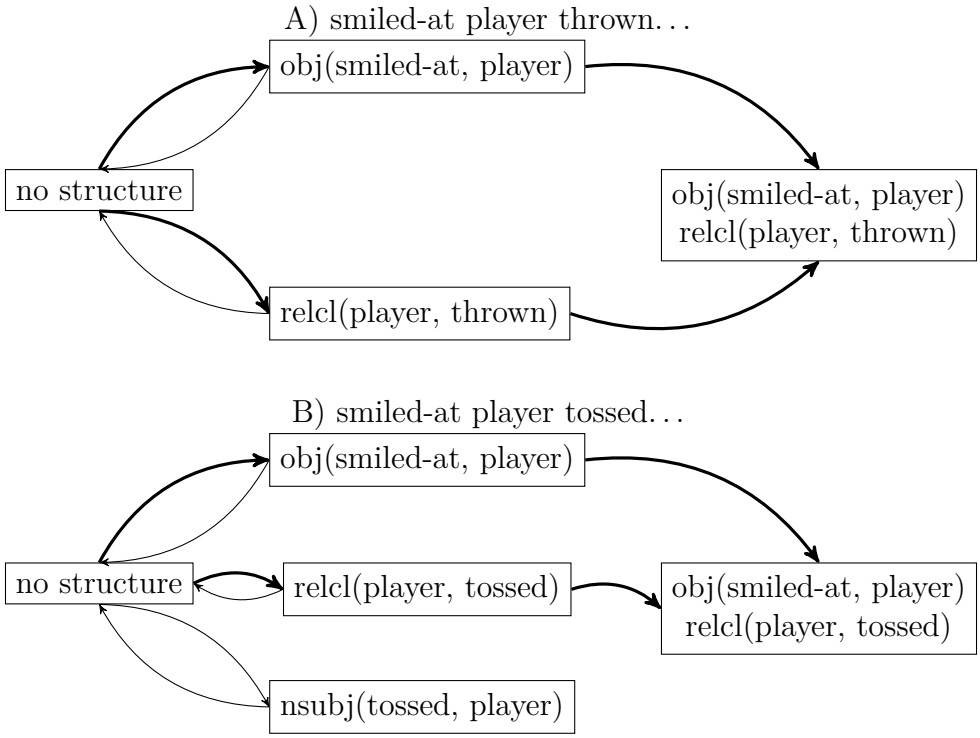

Figure 8: Network representation of the states for the reduced forms in (5). The arrows indicate possible transitions, with the thicker arrows indicating a higher transition rate than the thinner arrows. The upper network corresponds to (5) (b), the lower network to (5) (d).

the state with the ccomp(saw, had) link has a longer dependency, so its harmony is lower than the dead-end or nsubj states. This means that, for garden paths, the model is more likely to get garden pathed than it is to take one of the viable paths toward the absorbing state. As the noise parameter decreases, it becomes both harder to get out of the garden path state and harder to take a path to the absorbing state. This drives the average processing times to positive infinity. For slightly higher noise values, though, garden paths and local coherence effects arise in similar ways; the model sometimes gets temporarily stuck in a dead-end state that delays it from reaching the absorbing state and moving on to the next word.

The fact that mparse behaves unlike people in both garden paths (impossibly long processing times) and local coherence (flipped order of reduced and non-reduced processing times) suggests that very low noise settings might not correspond to realistic settings for modeling human behavior. For T values less than approximately two, the model effectively loses the ability to reanalyze in garden paths and build a viable parse in local coherence. This could just be a idiosyncrasy of the model, but it might also suggest that effective parsing requires some flexibility in parsing. We need to be able to backtrack through structural choices and press on through temporary difficulty if we want to comprehend sentences.

Finally, we highlight the fact that not all possible orderings of condition means are possible for the local coherence materials. The reduced locally coherent condition is always processed more slowly than the reduced non-locally coherent condition. As Roberts and Pashler (2000) argue, a model that can predict any possible ordering of conditions is not very informative about the process it purports to explain. Thus, while mparse does make unrealistic predictions for some parameter settings, it is still limited in the scope of effects it can explain.

3.3 The ambiguity advantage

Garden paths and local coherence effects are not ambiguous after the whole sentence has been read. Garden paths are only temporarily ambiguous, and locally coherent sentences only seem ambiguous when they are not. We have seen that mparse can explain how these effects can come about in a self-organization-based model, but it is also important to test the model on truly ambiguous sentences.

An intuitive prediction for globally ambiguous sentences is that the existence of multiple viable analyses should slow processing when compared to unambiguous sentences. However, Traxler et al. (1998) found the opposite effect in English in the structures in (6) (see also Swets, Desmet, Clifton, & Ferreira, 2008; van Gompel et al., 2005, 2001).

(6) a. The driver of the car that had the mustache was pretty cool.
b. The car of the driver that had the mustache was pretty cool.

c. The son of the driver that had the mustache was pretty cool.

They found a speedup in eye-tracking reading times when *that had the mustache* could plausibly attach to more than one noun phrase (6-c) compared to unambiguous high attachment in (6-a) and the unambiguous low attachment in (6-b). These findings have been argued to provide evidence against the idea that ambiguity automatically leads to processing difficulty. The dependency parses for these materials are shown in Fig. 9.

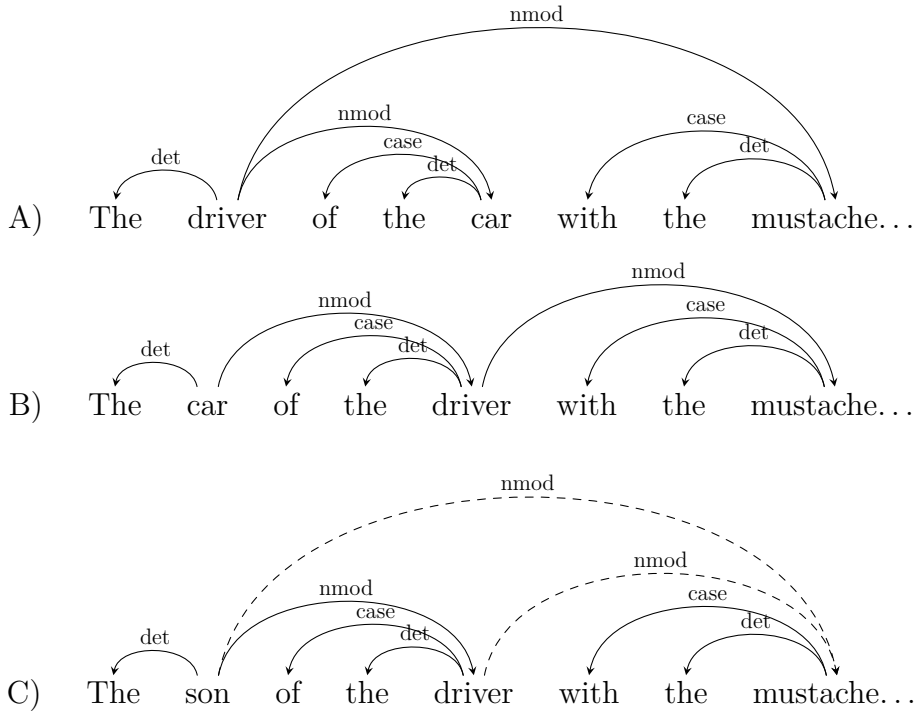

Figure 9: Dependency structures for materials for the ambiguity advantage. (A) shows the high-attachment preference, (B) the low-attachment preference, and (C) the globally ambiguous case. The dashed dependency links in (C) indicate that either is viable, but only one is possible at a time. Note that in the Universal Dependencies, prepositions are dependents (marked "case") of the nouns they are associated with.

The unrestricted race model (URM) of van Gompel et al. (2000) provides an account for the basic finding. In the URM, the parser starts building all (fully grammatical) syntactic structures compatible with the input. Whichever structure is completed first is the one adopted, and the parser moves on. If the parser builds a structure that turns out to be incompatible with subsequent input, it must reanalyze

the previously built structure, causing a processing slowdown. Under the URM, (6-c) is faster to process than (6-a) and (6-b) because wherever the prepositional phrase beginning with *with* attaches (to *son* or *driver*), it is compatible with the rest of the sentence, and so no reanalysis is ever necessary. In (6-a) and (6-b), however, sometimes the *with*-phrase will attach to an NP incompatible with the rest of the sentence, necessitating reanalysis and slowing reading times.

The ambiguity advantage is an interesting test case for mparse. Mean processing times in mparse reflect a complex interplay between the cost of having multiple reasonably well-formed structural alternatives and the exact structure of state network that mparse produces for a given sentence. Understanding how mparse processes globally ambiguous sentences will thus shed further light on the factors that affect mean processing times in the model.

3.3.1 Method

As for the other experiments, simplified materials were used for the ambiguity advantage in order to make the processing dynamics easier to understand and visualize (elided words in brackets):

(7) a. driver [of] car [with] mustache
b. car [of] driver [with] mustache
c. son [of] driver [with] mustache

Word-by-word model output for the full sentences in (6) is provided in Appendix B. Mean reading times from mparse were calculated using the grammar shown in Table 1.

3.3.2 Results and discussion

The mean processing times at the final words in (7) are plotted in the top panel of Fig. 10. Similar to the pattern in total reading times (eye-tracking) in Traxler et al. (1998), mean processing times for the globally ambiguous condition are faster than the high-attachment and low-attachment conditions. In addition, mparse predicts that the high-attachment condition should be read more slowly than the low-attachment condition. This is because the high-attachment condition involves a longer-distance dependency between *driver* and its dependent *mustache*. This lowers the overall harmony structures that contain that dependency compared to the low-attachment condition, where the same dependency spans fewer words. This fits with the pattern observed in Swets et al. (2008) with items with no or superficial comprehension questions.

Fig. 11 provides more information about how these results arise. Fig. 11 (A) shows the state network for the high-attachment preference (7-a). In contrast

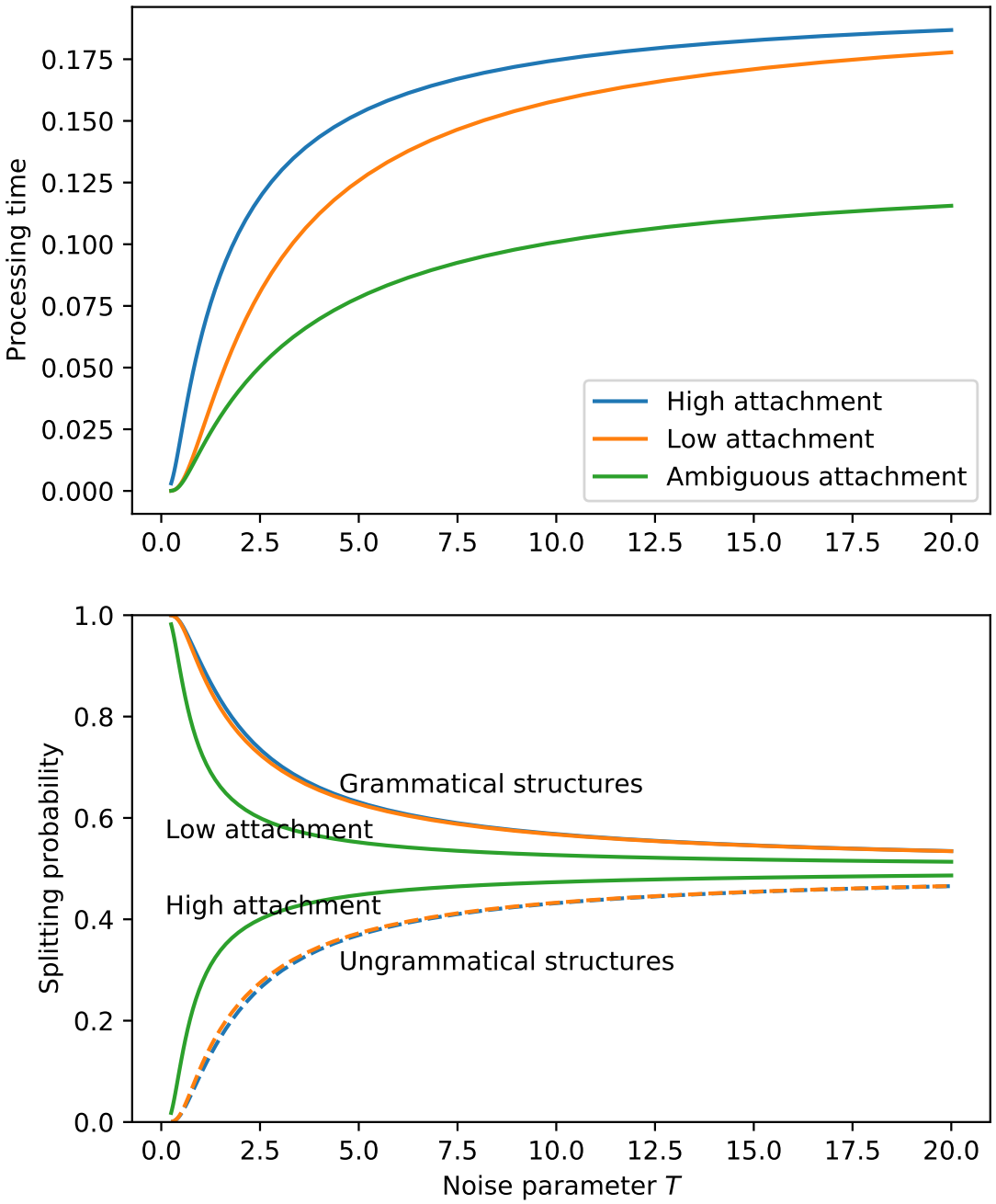

Figure 10: Top panel: Mean processing times by condition. Bottom panel: splitting probabilities. The dashed curves marked "ungrammatical" are parses where the first noun attaches as the nominal modifier of the second, in violation of the word order preference in the grammar. For the globally ambiguous condition in green, both absorbing states are grammatical, although the higher harmony of the low-attachment structure leads to a higher splitting probability.

to the garden path and local coherence examples, there are now two absorbing states because the all three sentences are globally ambiguous according to mparse's grammar. One absorbing state corresponds to the the fully grammatical, correct parse where *mustache* and *car* are modifiers of *driver*. The other absorbing state reverses the dependency between *car* and *driver*, with *car* now taking *driver* as its modifier. This violates the word order preference of the nominal modifier rule of the grammar, so its overall harmony is lower. The relatively low harmony leads to lower splitting probability for that state, shown in the lower panel of Fig. 10. The splitting probabilities are the probabilities that mparse will end up in different absorbing states.[5] The state network for the low-attachment condition (7-b) is similar (with an ungrammatical state where both *car* and *mustache* attach as nominal modifiers of *driver*); this leads to similar splitting probabilities.

This illustrates an important property of mparse: Coding the grammar in binary relations between pairs of words can lead to unexpected and ill-formed structures being considered. However, unless the noise parameter T is set to a high value, those states will play only a minor role in parsing, and grammatical structures will be built with a high probability. The mathematical formalism behind mparse allows us to explicitly calculate those probabilities and quantify just how strong an influence extra-grammatical parses can exert.

For the globally ambiguous condition (7-c), both absorbing states (Fig. 11 (B)) have high harmony. The low-attachment structure (where *mustache* attaches as the dependent of *driver* instead of *son*) has a slightly higher harmony due to its shorter dependency lengths, which is also reflected in the slightly higher splitting probability (Fig. 10, green lines in the lower panel).

These results show that mparse explains the ambiguity effect in a similar way to the URM. In mparse, multiple paths to high-harmony absorbing states lead to fast processing times, just as a race process between two well-formed parses in the URM does. Mparse and the URM are also similar in that both end up with a single parse at the end of processing a word. Arriving at different parses takes different amounts of time (in general), and so both models predict different processing time distributions depending on which parse is built. The URM has been implemented computationally in Logačev and Vasishth (2015) and Logačev and Vasishth (2016), so future work can compare mparse's predictions with those of Logačev and Vasishth's models. We note, though, that Logačev and Vasishth's models were meant to capture task effects first reported by Swets et al. (2008), where the ambiguity advantage effect can be attenuated when post-sentence comprehension questions ask about the ambiguous attachment. Future work must determine how to incorporate such task effects into mparse. For now, we must be

[5]When there is only one absorbing state (like for garden path and local coherence examples), probability of absorption into that state is one.

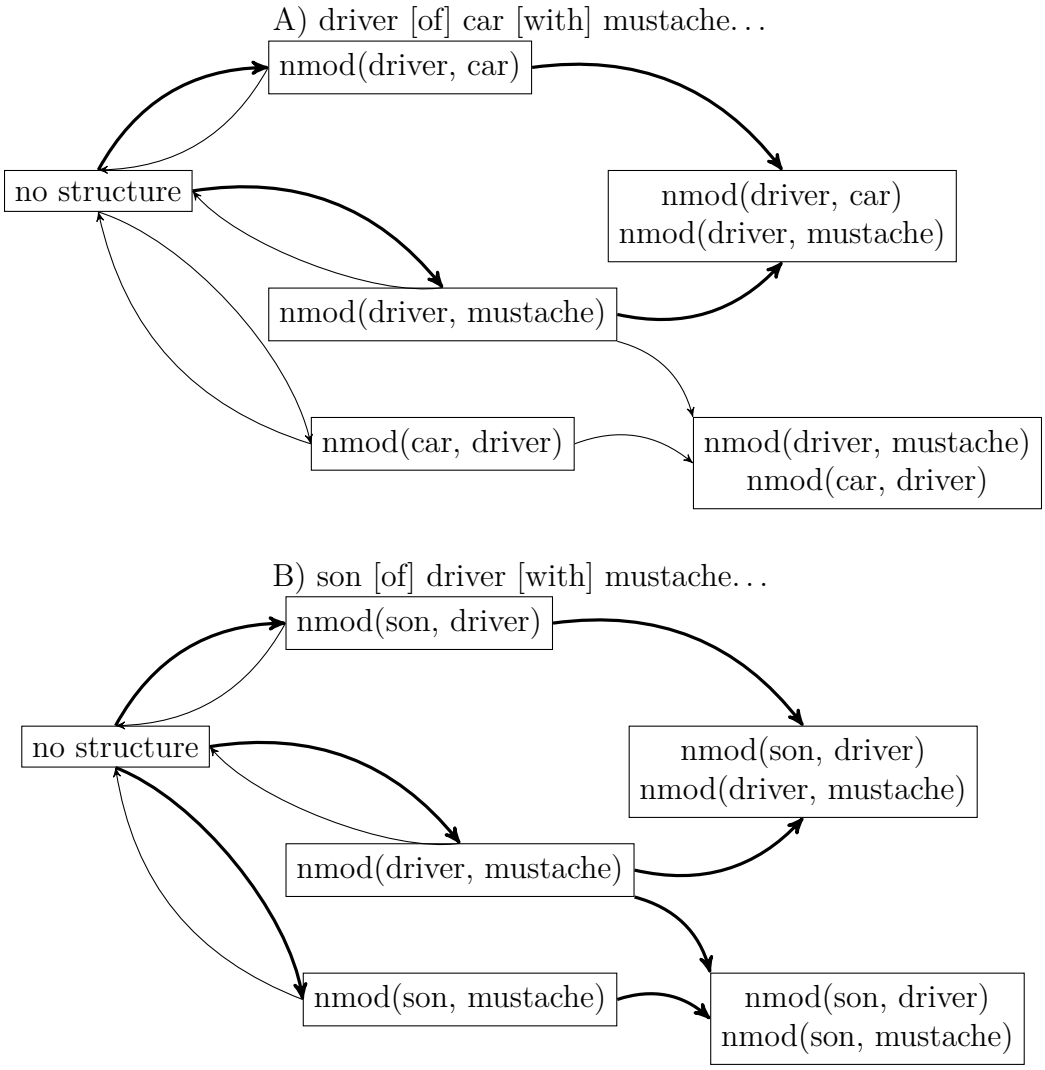

Figure 11: Network representation of the states for the simplified forms in (7). The arrows indicate possible transitions, with the thicker arrows indicating a higher transition rate than the thinner arrows. The top network corresponds to (7) (a) and the lower network to (7) (c).

content with mparse's ability to reproduce the basic effect of Traxler et al. (1998).

It is also important to note that the low-attachment preference typically seen in English does not hold in all other languages, and other orderings of the high, low, and ambiguous attachment conditions have been observed. For example, Chernova and Chernigovskaya (2015) found a high attachment preference in offline measures and no ambiguity advantage in self-paced reading and eye tracking in Russian. Cuetos and Mitchell (1988) report a similar high-attachment preference in Spanish. Thus, future cross-linguistic work with mparse might require us to rethink the inclusion or the weighting of the penalty for longer dependencies, as the current implementation assumes longer links always result in lower-harmony structures. This assumption might not hold for languages other than English.

4 General discussion

Mparse is a model of incremental sentence parsing in which dependency parses self-organize. This process is implemented using the master equation formalism borrowed from physics, chemistry, and biology (Iyer-Biswas & Zilman, 2016; Oppenheim et al., 1977; van Kampen, 2007). Analysis using the master equation formalism allows us to determine analytically how probable different structural analyses are and to make word-by-word reading time predictions.

We showed that mparse can account for local coherence effects (Tabor et al., 2004), the difference in effect size for two types of garden paths (Sturt et al., 1999), and the ambiguity advantage (Traxler et al., 1998) for a broad range of parameter settings. Only for very low noise levels, where the model effectively looses the ability to reanalyze, do the predicts fail to correspond to established human reading time effects. Local coherence and garden paths are caused when the model gets temporarily trapped in isolated "dead-end" parses. Mparse thus provides a unified analysis of two seemingly unrelated sentence comprehension effects. Mparse reproduces the ambiguity advantage because globally ambiguous structures offer multiple high-harmony paths to a fully harmonious absorbing state (similar to the URM; van Gompel et al., 2000), whereas the high- and low-attachment conditions contain some paths to low-harmony absorbing states that take longer to build.

Mparse improves upon previous self-organizing models in a number of ways. First, it contains only one truly free parameter that controls the noise (T; there is an additional scaling parameter that has no effect on the qualitative shape of processing time predictions. See Appendix A.). The processing time effects hold over a wide range of this noise parameter, so hand-picking a value to demonstrate results was not necessary. Second, mparse correctly predicts reading time patterns, in contrast to some (but not all) previous self-organizing models (e.g., Kempen & Vosse, 1989; Smith et al., 2018). Finally, mparse can reproduce local coherence

effects, two types of garden paths, and the ambiguity advantage all using the same set of parameters. In fact, for the simulations presented here, the model was created once with a single set of grammar rules and then tested with all constructions while varying the noise parameter simultaneously for all sentences. This suggests that mparse could scale up to broad-scale parsing (see future directions below).

A key innovation in mparse is the application of the master equation. This formalism describes the random-walk parsing process by saying how the probabilities of different parse states changes in time. The powerful analyses that this formalism allows (Iyer-Biswas & Zilman, 2016; Oppenheim et al., 1977; Polizzi, Therien, & Beratan, 2016; van Kampen, 2007) set mparse apart from many previous self-organization-based models, where the most viable way of understanding a model was to simulate it many times and see what happens (Kempen & Vosse, 1989; Smith et al., 2018, 2021; Smith & Tabor, 2018; Tabor & Hutchins, 2004; Vosse & Kempen, 2000, 2009). Indeed, couching mparse's dynamics in this well-understood mathematical framework allows us to make quantitative reading time predictions, an improvement over some previous self-organizing models (e.g., Kempen & Vosse, 1989; Smith et al., 2018). Only the gradient symbolic computation models, perhaps, can be as extensively analytically investigated as mparse to understand sentence processing effects, although these models do not allow ungrammatical structures to form[6] (Cho, Goldrick, Lewis, & Smolensky, 2018; Cho et al., 2017; Cho & Smolensky, 2016; Tupper, Smolensky, & Cho, 2018).

Mparse is close in spirit to information theoretic approaches to sentence processing (Hale, 2003, 2016; R. Levy, 2008a). These approaches have in common that changes in the probability distribution over structures have a causal effect on processing times. For entropy reduction (Hale, 2003), reading times are proportional to how much uncertainty has been reduced after reading a new word. Surprisal theory states that reading times are proportional to the change in the probability distribution over structures from one word to the next. Mparse explicitly models the change in the probabilities of different structures, so it will be possible in future work to calculate entropy and surprisal values at each word while mparse processes it. Entropy is a summary statistic of a probability distribution, and time-dependent summary statistics are simple to calculate using the master equation (e.g. Y. Levy, Jortner, & Berry, 2002; Lu, Zhang, & Berry, 2005). Surprisal for each word can be easily calculated as well from as the Kullback-Leibler divergence between the splitting probabilities at word w_n and word w_{n+1}. Thus, it is possible to make very detailed comparisons between these frameworks and possibly to derive diverging, testable predictions.

[6]Structural blends are possible, that is, superpositions of two or more grammatical structures. These blends do not resolve to globally incoherent parses, though (Cho, Goldrick, & Smolensky, 2017).

4.1 Limitations

Every computational model has limitations. We have tried to be as explicit as possible about the choices made in implementing mparse, but it is possible that the successes mparse shows in reproducing existing effects are due to particular choices. A different grammar formalism, for example could lead to differently structured state networks, which fundamentally affect processing time predictions. For example, the fact that the difference in NP/S and NP/Z garden path effects is driven by differences in the *control* conditions might stem from our choice of grammar.

The particular form of the harmony function is also a crucial choice. We chose to penalize parse states if they have long dependencies; this led to diverging processing time predictions with garden path materials but not with local coherence materials. Removing that constraint would likely lead to different garden path results and could cause there to be no difference between NP/S and NP/Z garden paths in mparse.

We have tried to justify our choices, but others are welcome to question them and try out different ones. All model data are generated using Python embedded in the Latex source file for this paper using Pweave (Pastell et al., 2017), so our results can be easily reproduced. The source code for mparse and the Latex source for this paper are available at `https://osf.io/k6rnx/`.

In this paper, we have focused on mean processing times as a function of the noise parameter. But as Roberts and Pashler (2000) discuss, the variability in the human data plays just as important a role in evaluating a model as the model's predictions do. If the human data are too noisy or variable, model fit is not very informative because we do not actually know much about how people perform. Future work with mparse should compare the range of model predictions to the range of effects from experiments, as was done in Vasishth, Mertzen, Jäger, and Gelman (2018) and Jäger, Mertzen, Van Dyke, and Vasishth (2019) for the cue-based retrieval model of Lewis and Vasishth (2005) in order to determine the extent to which mparse's predictions overlap with the range of reading times humans produce.

Finally, mparse does not implement any kind of prediction, at least not in the sense of prediction where structure or even particular lexical items are activated before they are encountered in the input. Examples of participants looking at referents of words in anticipation of words not yet encountered (e.g., Altmann & Kamide, 1999; Kukona, Cho, Magnuson, & Tabor, 2014) cannot be explained with the current form of mparse. Additional work is needed to determine whether prediction in this sense can be added to mparse.

4.2 Future directions

As mentioned above, this paper only scratches the surface of the mathematical details that can be extracted from mparse's master equation formalism. Among others, full processing time probability density functions conditional on the initial and absorbing states can be derived (Iyer-Biswas & Zilman, 2016; Polizzi et al., 2016; Valleriani, Li, & Kolomeisky, 2014; Valleriani, Liepelt, & Lipowsky, 2008). This will generate new predictions about what reading times at each word in a sentence should look like given which parse was built at the previous word and which parse is built at the current word. Also, moment-by-moment analyses of which transient states are causing processing slowdowns are possible (Lu et al., 2005; Miller, Doye, & Wales, 1999). These analyses could lead to new predictions for fixation trajectories in visual world eye-tracking experiments. Testing these new predictions will likely require new experimental methods and high-powered human experiments to test.

Currently, the link harmonies in mparse are set to one. This was done as a convenience, but more realistic processing time effects are likely possible if the link harmonies were calculated from large corpora. This would allow graded subcategorization preferences and other structural frequency effects to affect processing times. One approach way of estimating link harmonies was presented in Smith and Vasishth (2020). They created high-dimensional feature vectors for words and for their dependents from co-occurrence data in a parsed corpus. The cosine similarity between word vectors and dependent vectors can be used as a measure of link harmony. Importantly, their method is model-agnostic, so the same feature vectors can be used in mparse as in the cue-based retrieval model, for example (Engelmann, Jäger, & Vasishth, 2019; Lewis & Vasishth, 2005; Vasishth et al., 2019). This would make the models more comparable and would facilitate quantitative model comparison between very different approaches to sentence processing.

The dependency grammar rules that mparse uses can also be taken directly from parsed corpora, for example the gold-standard corpora available from the Universal Dependencies research group (Nivre et al., 2016). This would open the door to truly broad-coverage reading time predictions (testing against self-paced reading times in the Natural Stories Corpus, Futrell, Gibson, Tily, et al., 2020, for example) in English and a large number of other languages.

Future work should also investigate possible cognitive correlates of the noise parameter T. Small noise causes mparse to strongly prefer well-formed over ill-formed states, while large noise loosens the effect of harmony on the parsing choices the model makes. We speculate that one might manipulate the noise parameter by changing task demands. It might be that encouraging participants pay close attention to the structures they build (e.g., by asking detailed comprehension questions) might put them in a low-noise regime where they mostly build correct

parses but simultaneously exhibit enormous garden path effects in a small number of trials. Small noise also corresponded to smaller differences between conditions with the ambiguity advantage materials (see Fig. 10, top panel). This could potentially explain why Swets et al. (2008) did not find an ambiguity advantage when they asked comprehension questions about the ambiguous parts of the sentences (see also Logačev & Vasishth, 2015). Future work could test these predictions by varying the type of comprehension question participants receive and then fitting the T parameter to participant data.

Finally, the processing time predictions that mparse makes are not on the same scale as human reading times; they are given in arbitrary units. However, mparse contains a free scaling parameter (in addition to the free noise parameter T) that can be used to put its processing time predictions on the millisecond scale, similar to the latency factor parameter in the cue-based retrieval model. This scaling parameter can easily be fit to human reading time data in a Bayesian context using approximate Bayesian computation (Palestro, Sederberg, Osth, Van Zandt, & Turner, 2018; Sisson & Fan, 2019). This method provides a Bayesian credible interval for likely values of the parameter given the data. Even without estimating link harmonies from corpus data, this step will immediately allow quantitative model comparison to test different models on their fit to human data, for example, using the data sets on similarity based interference summarized in the Bayesian meta-analysis of Jäger et al. (2017).

4.3 Conclusion

Mparse is a new framework for exploring how human sentence parsing might self-organize through local, word-word interactions. This idea is not new in psycholinguistics, but previous implementations have not allowed as rich a mathematical analysis as mparse does. We hope that by implementing self-organization in a more tractable and less-ad-hoc way will encourage even more explicit theory building and new experiments to test the more detailed empirical predictions that are possible with richer theory.

References

Altmann, G. T. M., & Kamide, Y. (1999). Incremental interpretation at verbs: Restricting the domain of subsequent reference. *Cognition*, *73*(3), 247–264.

Bever, T. G. (1970). The cognitive basis for linguistic structures. In R. Hayes (Ed.), *Cognition and Language Development* (pp. 279–362). Wiley & Sons.

Bicknell, K., & Levy, R. (2009). A model of local coherence effects in human sentence processing as consequences of updates from bottom-up prior to

posterior beliefs. In *Proceedings of the 10th Annual Meeting of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) Conference* (pp. 665–673). Boulder, Colorado, USA.

Chernova, D., & Chernigovskaya, T. (2015). Syntactic ambiguity resolution in sentence processing: New evidence from a morphologically rich language. In G. Airenti, B. G. Bara, & G. Sandini (Eds.), *Proceedings of the EuroAsian-Pacific Joint Conference on Cognitive Science* (Vols. 129–133).

Cho, P. W., Goldrick, M., Lewis, R. L., & Smolensky, P. (2018). Dynamic encoding of structural uncertainty in gradient symbols. In *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics.*

Cho, P. W., Goldrick, M., & Smolensky, P. (2017). Incremental parsing in a continuous dynamical system: Sentence procesing in Gradient Symbolic Computation. *Linguistics Vanguard*, *3*(1).

Cho, P. W., & Smolensky, P. (2016). Bifurcation analysis of a gradient symbolic computation model of incremental processing. In A. Papafragou, D. Grodner, D. Mirman, & J. C. Trueswell (Eds.), *Proceedings of the 38th Annual Conference of the Cognititve Science Society* (pp. 1487–1492).

Chomsky, N. (1957/2002). *Syntactic structures* (2nd ed.). Berlin: Walter de Gruyter.

Christianson, K., Luke, S. G., Hussey, E. K., & Wochna, K. L. (2017). Why reread? Evidence from garden-path and local coherence structures. *The Quarterly Journal of Experimental Psychology*, *7*(7), 1380–1405. doi: 10.1080/17470218.2016.1186200

Cuetos, F., & Mitchell, D. C. (1988). Cross-linguistic differences in parsing: Restrictions on the use of the late closure strategy in Spanish. *Cognition*, *30*, 73–105.

de Marneffe, M.-C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., & Manning, C. D. (2014). Universal Stanford dependencies: A cross-linguistic typology. In *International Conference on Language Resources and Evaluation* (Vol. 14, pp. 4585–4592).

de Marneffe, M.-C., & Nivre, J. (2019). Dependency grammar. *Annual Review of Linguistics*, *5*(197–218). doi: 10.1146/annurev-linguistics-011718-011842

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*, 85–103.

Engbert, R., Longtin, A., & Kliegl, R. (2002). A dynamical model of saccade generation in reading based on spatially distributed lexical processing. *Vision Research*, *42*, 621–636.

Engbert, R., Nuthmann, A., Richter, E. M., & Kliegl, R. (2005). SWIFT: A

dynamical model of saccade generation during reading. *Psychological Review*, *112*(4), 777–813.

Engelmann, F., Jäger, L. A., & Vasishth, S. (2019). The effect of prominence and cue association in retrieval processes: A computational account. *Cognitive Science*, *43*(e12800).

Frazier, L., & Fodor, J. D. (1978). The sausage machine: A new two-stage parsing model. *Cognition*, *6*, 291–325.

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), e12814. doi: 10.1111/cogs.12814

Futrell, R., Gibson, E., Tily, H. J., Blank, I., Vishnevetsky, A., Piantadosi, S. T., & Fedorenko, E. (2020). The natural stories corpus: A reading-time corpus of english texts containing rare syntactic constructions. *Language Resources and Evaluation*. doi: 10.1007/s10579-020-09503-7

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). Valencia, Spain.

Gaifman, H. (1965). Dependency systems and phrase-structure systems. *Information and Control*, *8*, 304–337.

Gibson, E. (2006). The interaction of top-down and bottom-up statistics in the resolution of syntactic category ambiguity. *Journal of Memory and Language*, *54*(3), 363–388.

Glauber, R. (1963). Time-dependent statistics of the Ising model. *Journal of Mathematical Physics*, *4*(2), 294–307.

Grodner, D., Gibson, E., Argaman, V., & Babyonyshev, M. (2003). Against repair-based reanalysis in sentence comprehension. *Journal of Psycholinguistic Research*, *32*(2), 141–166.

Haag, G. (2017). *Modelling with the master equation*. Springer.

Haken, H. (1983). *Synergetics: An introduction* (3rd ed.). Springer-Verlag.

Hale, J. T. (2001). A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies* (pp. 1–8). Association for Computational Linguistics. doi: 10.3115/1073336.1073357

Hale, J. T. (2003). The information conveyed by words in sentences. *Journal of Psycholinguistic Research*, *32*(2), 101–123.

Hale, J. T. (2011). What a rational parser would do. *Cognitive Science*, *35*, 399–443. doi: 10.1111/j.1551-6709.2010.01145.x

Hale, J. T. (2016). Information-theoretical complexity metrics. *Language and Linguistics Compass*, *10*(9), 397–4122. doi: 10.1111/lnc3.12196

Hays, D. G. (1964). Dependency theory: A formalism and some observations. *Language*, *40*(4), 511–525.

Iyer-Biswas, S., & Zilman, A. (2016). First-passage processes in cellular biology. In S. A. Rice & A. R. Dinner (Eds.), *Advances in chemical physics* (Vol. 160, pp. 261–306). John Wiley & Sons.

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. doi: 10.1016/j.jml.2017.01.004

Jäger, L. A., Mertzen, D., Van Dyke, J. A., & Vasishth, S. (2019). *Interference patterns in subject-verb agreement and reflexives revisited: A large-sample study.*

Kamide, Y., & Kukona, A. (2018). The influence of globally ungrammatical local syntactic constraints on real-time sentence comprehension: Evidence from the visual world paradigm and reading. *Cognitive Science*, 1–23. doi: 10.1111/cogs.12694

Keller, F., Gunasekharan, S., Mayo, N., & Corley, M. (2009). Timing accuracy of web experiments: A case study using the webexp software package. *Behavior Research Methods*, *41*(1), 1–12. doi: 10.3758/BRM.41.1.12

Kempen, G., & Vosse, T. (1989). Incremental syntactic tree formation in human sentence processing: A cognitive architecture based on activation decay and simulated annealing. *Connection Science*, *1*(3), 273–290.

Kimball, J. (1973). Seven principles of surface structure parsing in natural language. *Cognition*, *2*(1), 15–47.

Konieczny, L. (2005). The psychological reality of local coherences in sentence processing. In *Proceedings of the 27th Annual Conference of the Cognitive Science Society* (pp. 1178–1183).

Konieczny, L., Müller, D., Hachmann, W., Schwarzkopf, S., & Wolfer, S. (2009). Local syntactic coherence interpretation. evidence from a visual world study. In *Proceedings of the 31st annual conference of the Cognitive Science Society.*

Kukona, A., Cho, P. W., Magnuson, J. S., & Tabor, W. (2014). Lexical interference effects in sentence processing: Evidence from the visual world paradigm and self-organizing models. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *40*(2), 326–347. doi: 10.1037/a0034903

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

Levy, R. (2008a). Expectation-based syntactic comprehension. *Cognition*, *106*(3), 1126–1177.

Levy, R. (2008b). A noisy-channel model of rational human sentence compre-

hension under uncertain input. In *Proceedings of the 2008 conference on empirical methods in natural language processing* (pp. 234–243). Association for Computational Linguistics.

Levy, R., Bicknell, K., Slattery, T., & Rayner, K. (2009). Eye movement evidence that readers maintain and act on uncertainty about past linguistic input. *Proceedings of the National Academy of Sciences*, *106*(50), 21086–21090.

Levy, Y., Jortner, J., & Berry, R. S. (2002). Eigenvalue spectrum of the master equation for hierarchical dynamics of complex systems. *Physical Chemistry Chemical Physics*, *4*, 5052–5058. doi: 10.1039/B203534K

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

Logačev, P., & Vasishth, S. (2015). A multiple-channel model of task-dependent ambiguity resolution in sentence comprehension. *Cognitive Science*, *40*(2), 1–33. doi: 10.1111/cogs.12228

Logačev, P., & Vasishth, S. (2016). Understanding underspecification: A comparison of two computational implementations. *The Quarterly Journal of Experimental Psychology*, *69*(5), 996–1012. doi: 10.1080/17470218.2015.1134602

Lu, J., Zhang, C., & Berry, R. S. (2005). Kinetics of model energy landscapes: An approach to complex systems. *Physical Chemistry Chemical Physics*, *7*, 3443–3456.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, *21*(6), 1087–1092. doi: 10.1063/1.1699114

Miller, M. A., Doye, J. P. K., & Wales, D. J. (1999). Structural relaxation in atomic clusters: Master equation dynamics. *Physical Review E*, *60*(4), 3701–3718.

Morgan, E., Keller, F., & Steedman, M. (2010). A bottom-up parsing model of local coherence effects. In *Proceedings of the 32nd annual meeting of the cognitive science society.*

Müller, H. M., & Konieczny, L. (2019). The effect of context on local syntactic coherency processing. In *Proceedings of the 32nd annual CUNY sentence processing conference.*

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C. D., ... Zeman, D. (2016). Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the tenth international conference on language resources and evaluation (lrec 2016).*

Oppenheim, I., Shuler, K. E., & Weiss, G. (1977). *Stochastic processes in chemical physics: The master equation.* MIT Press.

Oppenheim, I., Shuler, K. E., & Weiss, G. H. (1967). Stochastic theory of multistate relaxation processes. *Advances in Molecular Relaxation Processes*, *1*, 13–68.

Paape, D., & Vasishth, S. (2015). Local coherence and preemptive digging-in effects

in German. *Language and Speech*, 1–17. doi: 10.1177/0023830915608410

Palestro, J. J., Sederberg, P. B., Osth, A. F., Van Zandt, T., & Turner, B. M. (2018). *Likelihood-free methods for cognitive science*. Springer.

Park, S., Sener, M. K., Lu, D., & Schulten, K. (2003). Reaction paths based on mean first-passage times. *Journal of Chemical Physics*, *119*(3). doi: 10.1063/1.1570396

Pastell, M., abukaj, O'Leary, A., Laverde, R., van Foreest, N., stonebig, ... Vaillant, G. A. (2017, August). *mpastell/pweave: Pweave v0.30.* Zenodo. Retrieved from `https://doi.org/10.5281/zenodo.850887` doi: 10.5281/zenodo.850887

Pearlmutter, N. J., Garnsey, S. M., & Bock, K. (1999). Agreement processes in sentence comprehension. *Journal of Memory and Language*, *41*, 427–456.

Polizzi, N. F., Therien, M. J., & Beratan, D. N. (2016). Mean first-passage times in biology. *Israel Journal of Chemistry*, *56*, 816–824. doi: 10.1002/ijch.201600040

Prasad, G., & Linzen, T. (2019). *How much harder are hard garden path sentences than easy ones?* Retrieved from `https://osf.io/qczjk/` (Poster presented at the 41st Annual Conference of the Cognitive Science Society)

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

Sisson, S. A., & Fan, Y. (Eds.). (2019). *Handbook of approximate Bayesian computation*. CRC Press.

Smith, G., Franck, J., & Tabor, W. (2018). A self-organizing approach to subject-verb number agreement. *Cognitive Science*, *42*(S4), 1043–1074. doi: 10.1111/cogs.12591

Smith, G., Franck, J., & Tabor, W. (2021). Encoding interference effects support self-organized sentence processing. *Cognitive Psychology*, *124*(101356).

Smith, G., & Tabor, W. (2018). Toward a theory of timing effects in self-organized sentence processing. In I. Juvina, J. Houpt, & C. Myers (Eds.), *Proceedings of the 16th International Conference on Cognitive Modeling* (pp. 138–143). Madison, Wisconsin: University of Wisconsin.

Smith, G., & Vasishth, S. (2020). A principled approach to feature selection in models of sentence processing. *Cognitive Science*, *44*(12), e12918. doi: 10.1111/cogs.12918

Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. I: Foundations, pp. 194–281). MIT Press.

Sturt, P., & Crocker, M. W. (1997). Thematic monotonicity. *Journal of Psycholinguistic Research*, *26*(3), 297–322.

Sturt, P., Pickering, M. J., & Crocker, M. W. (1999). Structural change and reanalysis difficulty in language comprehension. *Journal of Memory and Language*, *40*, 136–150.

Swets, B., Desmet, T., Clifton, C. J., & Ferreira, F. (2008). Underspecification of syntactic ambiguities: Evidence from self-paced reading. *Memory & Cognition*, *36*(1), 201–216. doi: 10.3758/MC.36.1.201

Tabor, W., Galantucci, B., & Richardson, D. (2004). Effects of merely local syntactic coherence on sentence processing. *Journal of Memory and Language*, *50*(4), 355–370.

Tabor, W., & Hutchins, S. (2004). Evidence for self-organized sentence processing: digging-in effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *30*(2), 431–450.

Traxler, M. J., Pickering, M. J., & Clifton, C. J. (1998). Adjunct attachment is not a form of lexical ambiguity resolution. *Journal of Memory and Language*, *39*, 558–592.

Tupper, P., Smolensky, P., & Cho, P. W. (2018). *Discrete symbolic optimization and boltzmann sampling by continuous neural dynamics: Gradient symbolic computation.* Retrieved from `https://arxiv.org/abs/1801.03562` (arXiv preprint)

Valleriani, A., Li, X., & Kolomeisky, A. B. (2014). Unveiling the hidden structure of complex stochastic biochemical networks. *The Journal of Chemical Physics*, *140*(064101). doi: 10.1063/1.4863997

Valleriani, A., Liepelt, S., & Lipowsky, R. (2008). Dwell time distributions for kinesin's mechanical steps. *Europhysics Letters*, *82*(28011). doi: 10.1209/025-5075/82/28011

van Kampen, N. G. (2007). *Stochastic processes in physics and chemistry.* Elsevier.

van Gompel, R. P., Pickering, M. J., Pearson, J., & Liversedge, S. P. (2005). Evidence against competition during syntactic ambiguity resolution. *Journal of Memory and Language*, *52*, 284–307. doi: 10.1016/j.jml.2004.11.003

van Gompel, R. P., Pickering, M. J., & Traxler, M. J. (2000). Unrestricted race: A new model of syntactic ambiguity resolution. In A. Kennedy, R. Radach, D. Heller, & J. Pynte (Eds.), *Reading as a perceptual process* (pp. 621–648). Amsterdam: Elsevier.

van Gompel, R. P., Pickering, M. J., & Traxler, M. J. (2001). Reanalysis in sentence processing: Evidence against current constraint-based and two-stage models. *Journal of Memory and Language*, *45*, 225–258.

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), P968–982. doi: 10.1016/j.tics.2019.09.003

Vosse, T., & Kempen, G. (2000). Syntactic structure assembly in human parsing: a computational model based on competitive inhibition and a lexicalist grammar. *Cognition*, *75*, 105–143.

Vosse, T., & Kempen, G. (2009). The Unification Space implemented as a localist neural net: predictions and error-tolerance in a constraint-based parser. *Cognitive Neurodynamics*, *3*(4), 331-346.

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

Weidlich, W. (1991). Physics and social science—the approach of synergetics. *Physics Reports*, *204*(1), 1–163.

Weiss, G. H. (1966). First passage times in chemical physics. *Advances in Chemical Physics*, *13*(1), 1–18.

A Mathematical details

Notation Scalars are written in italics: x, y. Vectors are written in lowercase in bold, e.g., $\mathbf{p}$, and are assumed to be column vectors unless they are transposed to row vectors using the $^\intercal$ operator, e.g., $\mathbf{p}^\intercal$. Matrices are written in uppercase in boldface: $\mathbf{A}$. Elements of vectors or matrices are written as p_n or A_{ij}. The cardinality of a set S is denoted by $|S|$.

A.1 Calculating harmony

The equation for the harmony of a state is given in Eq. 1:

$$\begin{aligned} h = & \left[\sum_l \text{harmony}(l)\right] \\ & - \left[|n_{\text{links}} - (w-1)|\right] \\ & - \left[\sum_l \text{order preference}(l) * \text{sgn}(\text{word nr}(\text{dep}(l)) \right. \\ & \qquad \left. - \text{word nr}(\text{head}(l)))\right] \\ & - \left[\sum_l \text{word nr}(\text{dep}(l)) - \text{word nr}(\text{head}(l))\right] \end{aligned} \tag{1}$$

The notation $\sum_l$ denotes the sum over all links l in a state. The first line of Eq. 1 sums the harmonies of each link. The second line is the penalty for having too

many or too few links, i.e., the number of links $n_{\text{links}} \neq w - 1$. The third line decrements the state's harmony by one when the if the difference between the linear position word $\text{nr}(\cdot)$ of the dependent word of the l-th word $\text{dep}(\cdot)$ has a different sign ($\text{sgn}(\cdot)$) than the link's preferred word order (order preference$(\cdot)$). When linear ordering of the head and dependent matches the link's preference, the harmony is incremented by one. Finally, the last line of Eq. 1 penalizes long links by decreasing the state's harmony by the linear distance between the head and the dependent.

A.2 The master equation

The master equation is a set of coupled ordinary differential equations that describe how the probabilities of being in different states changes as a function of time. The probability of being in configuration i increases due to jumps to that configuration from other configurations j:

$$\text{rate into } i = \sum_{j \neq i}^{n} A_{ij} p_j(t),$$

where A_{ij} is the transition rate from configuration j to configuration i per unit time. At the same time, probability shifts away from i to other states j:

$$\text{rate out of } i = -p_i(t) \sum_{j \neq i}^{n} A_{ji}$$

These processes happen simultaneously; one can imagine probability flowing like a liquid between different states at different rates depending on the relevant A_{ij}. Combining the two rate terms, we arrive at the master equation:

$$\frac{\mathrm{d}p_i(t)}{\mathrm{d}t} = \sum_{j \neq i}^{n} A_{ij} p_j(t), -p_i(t) \sum_{j \neq i}^{n} A_{ji} \tag{2}$$

Eq. 2 can be written more compactly in matrix/vector form:

$$\frac{\mathrm{d}\mathbf{p}(\mathbf{t})}{\mathrm{d}t} = \mathbf{A}\mathbf{p}(t), \tag{3}$$

where $\mathbf{p}(t)$ is a column vector of the n state probabilities at time t, and $\mathbf{A}$ is an $n \times n$ matrix with the transition rates A_{ij}. To ensure that probability is conserved ($\sum_i^n p_i(t) = 1 \quad \forall t$), we set the diagonal terms of $\mathbf{A}$ to be the sum of the off-diagonal columns of $\mathbf{A}$: $A_{ii} = -\sum_{j \neq i} A_{ji}$. Thus, probability flowing to a configuration is balanced by probability flowing away from it. This property, combined with an initial state $p(0)$ that is a probability distribution ($\sum_i p_i(0) = 1$), guarantees that

the state of the system will remain a probability distribution for all time (Haken, 1983; Oppenheim et al., 1977; van Kampen, 2007). An exception is the diagonal terms for absorbing states; these are set to zero, which makes flow away from an absorbing state impossible.

We set the transition rates A_{ij} using an exponential function of the difference in harmonies between state j and state i:

$$A_{ij} = n\tau \exp\left(\frac{h_i - h_j}{T}\right) \tag{4}$$

As Haag (2017) and Weidlich (1991) note, the exponential function is the simplest functional form we can assume that meets the assumptions for transition rates: The A_{ij} must be greater than or equal to zero; they change monotonically with the difference $h_i - h_j$; and A_{ij} should be greater than A_{ji} if $h_i > h_j$. We have also tested other transition rate functions like the sigmoidal function of Glauber (1963) and capped exponential of Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller (1953); the results are qualitatively identical for reasonable levels of the noise parameter T.

The free rate parameter $\tau > 0$ determines how long it takes to make a single transition from one state to another; it has units of state transitions per unit time. If we assume seconds are the unit of time used, then the mean exit times produced are also given in seconds. Because the number of states changes as new words are added, τ is multiplied by n, the number of states in the system. This ensures that reading time predictions per word remain on the same time scale; without rescaling by the number of states, the reading time predictions increase with the number of words in the sentence because there are more states to explore. Overall, the A_{ij} have units of n state transitions per second, which can be thought of as a measure of parsing efficiency. Here, we set τ to 1.0, but future work will estimate τ from human reading times so that the model's predictions will be directly comparable with reading time experiments.

The solution to the master equation in Eq. 3 is given in Eq. 5:

$$\mathbf{p}(t) = e^{\mathbf{A}t}\mathbf{p}(0) \tag{5}$$

The vector $\mathbf{p}(0)$ is the initial probability distribution over states at time $t = 0$ (see below). The exponential of a matrix is defined as

$$e^{\mathbf{A}t} = \sum_{l=0}^{\infty} \frac{t^l}{l!}\mathbf{A}^l$$

Eq. 5 gives the probabilities of being in any given state at time t given that the system was initialized at $\mathbf{p}(0)$. This solution will be used below to derive reading time predictions. First, we discuss the transition rate matrix in more detail, though.

It is important to note that the dynamics described by the master equation are Markovian, that is, the decision on which state to jump to next depends only on the current state and not on previous ones. That does not imply that the grammatical structures that mparse explores are created by a Markov model, which is seen as an insufficient description of human language structure at least since Chomsky (1957/2002). The grammar mparse uses is equivalent to a context-free grammar (Gaifman, 1965). Mparse simply takes a Markovian random walk through structures generated by a more powerful grammar. Note that the master equation is not limited to finite or even countable state space. That means that the same equations can be used even if the grammar generates an infinite number of different parses for finite number. Solving the master equation is much harder in that case, though.

A.3 The structure of the transition rate matrix

A main assumption of mparse is that it can only jump between states that differ by a single dependency link. That is, it can add a link or remove one, but no other options are possible. This is reflected in the structure of the transition rate matrix $\mathbf{A}$, which only has non-zero entries A_{ij} where the states i and j differ only by a single dependency link. This restriction is reminescent of the parsing algorithm of Hale (2011), which, in contrast to mparse, only explores fully grammatical parse states.

Creating the transition rate matrix for the first word of a sentence is a special case. The only state possible is the no-structure state. But with only a single state, there is nowhere for mparse to transition from into that one state. Therefore, an additional dummy state is included in the state space for the first word. It exists solely for its probability to drain into the one actual state, the no-structure structure. The dummy state is given a harmony of negative infinity, so mparse will always transition to the no-structure structure. This makes the dynamics very simple at the first word, and reading time predictions will be identical for the first word of any sentence.

After the first word, the dummy state is removed, and new states are added based on the structural affordances of the new word as described in the last section. Thus, the transition rate matrix $\mathbf{A}$ is updated when a new word is input by adding new dimensions corresponding to transitions between newly added states. After the first word, there is states are not removed from the state space because partial or complete parses at word w remain at least partial parses at word $w + 1$. The transition rates between state do change, however, even between states that were already present because the number of states has increase, which affects the transition rates calculated according to Eq. 4. The transition rate matrix remains unchanged during the processing of a word, though; it only changes when a new

word is input.

A.4 When to stop processing a word: Absorbing states

As described in the main text, mparse stops processing a word once it reaches any state that contains the maximum number of dependency links possible given the words so far. In most cases, this will be a state with $w - 1$ links after reading w words. We say that these states are *absorbing states*: once the system enters one, it cannot transition to any other state. Instead, mparse inputs the next word, updates its transition rate matrix, and resumes jumping among states.

We implement this in the master equation formalism as follows. Let S be the set of absorbing states. Let the set T be set of transient states, states that are not absorbing. To make sure that mparse cannot transition away from any of the states s in S to any other state i, we set the transition rates A_{is} to zero. The transition rates *to* the states in S are calculated as usual.

We can now partition the full transition rate matrix $\mathbf{A}$ into submatrices:

$$\mathbf{A} = \begin{bmatrix} \mathbf{T} & \mathbf{0}_{|T|\times|S|} \\ \mathbf{S} & \mathbf{0}_{|S|\times|S|} \end{bmatrix} \tag{6}$$

Here, $\mathbf{T}$ is the $|T| \times |T|$ submatrix of $\mathbf{A}$ that contains only transitions among the transient states; $\mathbf{S}$ contains the transition rates from the transient states into the absorbing states (dimensions: $|S| \times |T|$). The two $\mathbf{0}$ matrices contain only zeros and have the dimensions given by their subscripts. Below, we will use these submatrices to derive reading time predictions at each word. But first, we need to determine the initial conditions at each word, i.e., where in the state space mparse is initialized to when it inputs a new word.

A.5 Initial conditions at each word

The transition rate matrix $\mathbf{A}$ tells us how mparse moves from one state to another at each time point. But in order to solve the master equation or calculate mean processing times, we have to specify the initial conditions, i.e., the vector $\mathbf{p}(0)$ of probabilities at time $t = 0$.

When only the first word has been read, there are two states, the dummy state, which does not correspond to any parse, and the no-structure state. In order to allow mparse to "build" the no-structure parse, all of the probability is initialized in the dummy state, i.e., $p_{\text{dummy}}(0) = 1.0, p_{\text{no-structure}}(0) = 0.0$ or $\mathbf{p}(0) = [1.0, 0.0]^\intercal$. Thus, the probability of being in the dummy state at time 0 is 1.0. Since the no-structure state is the most complete parse possible after reading a single word, it is the absorbing state. Mparse therefore processes the first word until it jumps

from the dummy state to the no-structure state. Once it does so, it inputs the next word, updates its transition rate matrix.

The initial conditions at each subsequent word are determined by the *splitting probabilities* at the preceding word. The splitting probabilities are the probabilities of being absorbed in a particular absorbing state without getting absorbed in another absorbing state first (Iyer-Biswas & Zilman, 2016; Park, Sener, Lu, & Schulten, 2003; Polizzi et al., 2016; Valleriani et al., 2014; van Kampen, 2007). The motivation for this is that, over many trials, mparse will be absorbed into the absorbing states at the rates given by the splitting probabilities. When a new word is input, mparse begins exploring the newly expanded state space starting from where it last was when it finished processing the previous word. Using the splitting probabilities as the initial conditions at the next word allows the final state at the previous word to affect processing at the next word.

Valleriani et al. (2014) show that the vector of splitting probabilities, given that the system started at $\mathbf{p}(0)$, is given by Eq. 7:

$$\mathbf{p}_{s\in S} = -\mathbf{S}\mathbf{T}^{-1}\mathbf{p}_{i\in T}(0) \tag{7}$$

The term $\mathbf{p}_{i\in T}(0)$ is vector of initial values for the states in the set of transient states.

For the first word in a sentence, the only absorbing state is the no-structure state, so the vector of splitting probabilities is just a scalar $p_{\text{no-structure}}$. The probability of absorption into this state given that mparse was initialized in the dummy state is one because probability only flows from the dummy state into the no-structure state. Thus, when the second word is input, the initial state is set to one for the no-structure state and zero otherwise.

For each word after that, the initial probability vector is set so that probabilities of previously absorbing states are equal to their splitting probabilities, and the probabilities of all other states are set to zero. A special case arises when the state space does not change from word w to word $w + 1$. This can happen when the newly added word cannot be attached in any way to the preceding words. In this case, the most complete states are ones that were already available, and mparse has already found one of them. But because the harmonies of the states has changed due to the introduction of a new word, mparse resets itself and uses a uniform distribution over the transient states, which are unchanged from word w, as the initial distribution $\mathbf{p}(0)$ at word $w + 1$.

We now have all we need to use mparse to make word-by-word reading time predictions. Given a pre-specified dependency grammar, mparse sets up a set of states at each word. It jumps probabilistically between them using the master equation (Eq. 3), with the rate of transitions between states per unit time determined by the difference in harmonies and Eq. **??**. Mparse jumps around among its transient

states until it finds an absorbing state, at which time the next word is input and the process starts again. The amount of time it takes to find an absorbing state can be calculated explicitly, without ever running the random walk. The next section shows how that works.

A.6 Predicting word-by-word reading times

As mentioned above, reading times in mparse are modeled by how long it takes for the model to reach a state with the maximum allowable number of attachment links given the words so far. We model this formally as an exit time problem (Oppenheim et al., 1977; Oppenheim, Shuler, & Weiss, 1967; van Kampen, 2007; Weiss, 1966): The exit time is defined as the amount of time it takes for a system to reach a set of absorbing states given that it started in a non-absorbing state, i.e., how long it takes for the system to exit the set of states it started in and enter one of the absorbing states.

In this paper, we focus on predictions of the mean of the exit time distribution. This allows us to check how well the predictions of mparse qualitatively match human reading time patterns. Oppenheim et al. (1977) derive a formula for the m-th non-central moment of the exit time distribution:

$$\mu_m = (-1)^m \, m! \, \mathbf{1}^\intercal \, \mathbf{T}^{-m} \, \mathbf{p}_{i \in T}(0) \tag{8}$$

The mean reading time at a word is then given by μ_1 and its variance by $\mu_2 - \mu_1^2$. Using Eq. 8, we can generate reading time predictions without ever running stochastic simulations; we can simply use the well-understood math behind the master equation formulism to calculate how stochastic simulations of mparse jumping between parse states would behave.

Note that while we focus on the mean here, far more detailed predictions are possible. We can actually derive explicit formulas for the full probability distribution of exit times, not just its moments. In fact, we can even derive probability distributions for reading times that are conditional on mparse starting in a particular initial state and being absorbed in a particular absorbing state, leading to $|T| \times |S|$ different reading time distributions at each word of the sentence. Deriving these distributions is straightforward (Iyer-Biswas & Zilman, 2016; Polizzi et al., 2016; van Kampen, 2007). The goal of the present paper, though, is just to introduce mparse as a theory of sentence comprehension. We will therefore discuss these additional details in a future paper.

B Word-by-word processing output

Shown below is the word-by-word output generated by mparse. For each new word in the string so far (shown in brackets), mparse calculates the mean and variance of the exit time distribution. For the simulations below, τ was set to 1.0, and T was set to 5.0. Note that some additional rules were added to the grammar to make parsing the full sentences possible. For the ambiguity advantage items, in particular, rules were added so that *son* and *driver* could each take the other as a nominal dependent. This was necessary to ensure that the number of states and paths to the absorbing states was approximately constant in all three conditions. The symmetrical link possibilities were not added in the main text for clarity; however, the qualitative pattern of results does not change. The globally ambiguous condition simply gets two additional absorbing states that are ungrammatical because of word order violations, similar to the high and low attachment conditions.

For each word in a sentence, the mean and variance of the processing time distribution is printed.

Garden paths

```
NP/S garden path control:
['the']
        Mean: 0.5, Var: 0.25
['the', 'woman']
        Mean: 0.335, Var: 0.112
['the', 'woman', 'saw']
        Mean: 0.205, Var: 0.048
['the', 'woman', 'saw', 'that']
        Mean: 0.233, Var: 0.052
['the', 'woman', 'saw', 'that', 'the']
        Mean: 0.131, Var: 0.017
['the', 'woman', 'saw', 'that', 'the', 'doctor']
        Mean: 0.091, Var: 0.007
['the', 'woman', 'saw', 'that', 'the', 'doctor', 'had']
        Mean: 0.062, Var: 0.003
NP/S garden path:
['the']
        Mean: 0.5, Var: 0.25
['the', 'woman']
        Mean: 0.335, Var: 0.112
['the', 'woman', 'saw']
```

```
        Mean: 0.205, Var: 0.048
['the', 'woman', 'saw', 'the']
        Mean: 0.127, Var: 0.016
['the', 'woman', 'saw', 'the', 'doctor']
        Mean: 0.078, Var: 0.005
['the', 'woman', 'saw', 'the', 'doctor', 'had']
        Mean: 0.083, Var: 0.004
NP/Z garden path control:
['before']
        Mean: 0.5, Var: 0.25
['before', 'the']
        Mean: 0.611, Var: 0.373
['before', 'the', 'woman']
        Mean: 0.335, Var: 0.112
['before', 'the', 'woman', 'visited,']
        Mean: 0.233, Var: 0.043
['before', 'the', 'woman', 'visited,', 'the']
        Mean: 0.121, Var: 0.015
['before', 'the', 'woman', 'visited,', 'the', 'doctor']
        Mean: 0.066, Var: 0.006
['before', 'the', 'woman', 'visited,', 'the', 'doctor', 'had']
        Mean: 0.043, Var: 0.002
NP/Z garden path:
['before']
        Mean: 0.5, Var: 0.25
['before', 'the']
        Mean: 0.611, Var: 0.373
['before', 'the', 'woman']
        Mean: 0.335, Var: 0.112
['before', 'the', 'woman', 'visited']
        Mean: 0.205, Var: 0.048
['before', 'the', 'woman', 'visited', 'the']
        Mean: 0.127, Var: 0.016
['before', 'the', 'woman', 'visited', 'the', 'doctor']
        Mean: 0.078, Var: 0.005
['before', 'the', 'woman', 'visited', 'the', 'doctor', 'had']
        Mean: 0.083, Var: 0.004
```

Local coherence effects

```
Locally coherent, reduced:
['the']
        Mean: 0.5, Var: 0.25
['the', 'coach']
        Mean: 0.335, Var: 0.112
['the', 'coach', 'smiled-at']
        Mean: 0.205, Var: 0.048
['the', 'coach', 'smiled-at', 'the']
        Mean: 0.127, Var: 0.016
['the', 'coach', 'smiled-at', 'the', 'player']
        Mean: 0.096, Var: 0.01
['the', 'coach', 'smiled-at', 'the', 'player', 'tossed']
        Mean: 0.063, Var: 0.009
['the', 'coach', 'smiled-at', 'the', 'player', 'tossed', 'the']
        Mean: 0.074, Var: 0.007
['the', 'coach', 'smiled-at', 'the', 'player', 'tossed', 'the',
'frisbee']
        Mean: 0.014, Var: 0.0

Locally coherent, non-reduced:
['the']
        Mean: 0.5, Var: 0.25
['the', 'coach']
        Mean: 0.335, Var: 0.112
['the', 'coach', 'smiled-at']
        Mean: 0.205, Var: 0.048
['the', 'coach', 'smiled-at', 'the']
        Mean: 0.127, Var: 0.016
['the', 'coach', 'smiled-at', 'the', 'player']
        Mean: 0.096, Var: 0.01
['the', 'coach', 'smiled-at', 'the', 'player', 'who']
        Mean: 0.104, Var: 0.011
['the', 'coach', 'smiled-at', 'the', 'player', 'who', 'was']
        Mean: 0.104, Var: 0.011
['the', 'coach', 'smiled-at', 'the', 'player', 'who', 'was', 'tossed']
        Mean: 0.045, Var: 0.002
['the', 'coach', 'smiled-at', 'the', 'player', 'who', 'was', 'tossed',
'the']
        Mean: 0.032, Var: 0.001
['the', 'coach', 'smiled-at', 'the', 'player', 'who', 'was', 'tossed',
```

```
'the', 'frisbee']
        Mean: 0.008, Var: 0.0

Non-locally coherent, reduced:
['the']
        Mean: 0.5, Var: 0.25
['the', 'coach']
        Mean: 0.335, Var: 0.112
['the', 'coach', 'smiled-at']
        Mean: 0.205, Var: 0.048
['the', 'coach', 'smiled-at', 'the']
        Mean: 0.127, Var: 0.016
['the', 'coach', 'smiled-at', 'the', 'player']
        Mean: 0.096, Var: 0.01
['the', 'coach', 'smiled-at', 'the', 'player', 'thrown']
        Mean: 0.055, Var: 0.004
['the', 'coach', 'smiled-at', 'the', 'player', 'thrown', 'the']
        Mean: 0.051, Var: 0.002
['the', 'coach', 'smiled-at', 'the', 'player', 'thrown', 'the',
'frisbee']
        Mean: 0.014, Var: 0.0

Non-locally coherent, non-reduced:
['the']
        Mean: 0.5, Var: 0.25
['the', 'coach']
        Mean: 0.335, Var: 0.112
['the', 'coach', 'smiled-at']
        Mean: 0.205, Var: 0.048
['the', 'coach', 'smiled-at', 'the']
        Mean: 0.127, Var: 0.016
['the', 'coach', 'smiled-at', 'the', 'player']
        Mean: 0.096, Var: 0.01
['the', 'coach', 'smiled-at', 'the', 'player', 'who']
        Mean: 0.104, Var: 0.011
['the', 'coach', 'smiled-at', 'the', 'player', 'who', 'was']
        Mean: 0.104, Var: 0.011
['the', 'coach', 'smiled-at', 'the', 'player', 'who', 'was', 'thrown']
        Mean: 0.046, Var: 0.002
['the', 'coach', 'smiled-at', 'the', 'player', 'who', 'was', 'thrown',
```

```
'the']
        Mean: 0.032, Var: 0.001
['the', 'coach', 'smiled-at', 'the', 'player', 'who', 'was', 'thrown',
'the', 'frisbee']
        Mean: 0.008, Var: 0.0
```

The ambiguity advantage

```
High attachment:
['the']
        Mean: 0.5, Var: 0.25
['the', 'driver']
        Mean: 0.335, Var: 0.112
['the', 'driver', 'of']
        Mean: 0.304, Var: 0.101
['the', 'driver', 'of', 'the']
        Mean: 0.179, Var: 0.035
['the', 'driver', 'of', 'the', 'car']
        Mean: 0.03, Var: 0.001
['the', 'driver', 'of', 'the', 'car', 'with']
        Mean: 0.027, Var: 0.001
['the', 'driver', 'of', 'the', 'car', 'with', 'the']
        Mean: 0.011, Var: 0.0
['the', 'driver', 'of', 'the', 'car', 'with', 'the', 'mustache']
        Mean: 0.015, Var: 0.0
Low attachment:
['the']
        Mean: 0.5, Var: 0.25
['the', 'car']
        Mean: 0.335, Var: 0.112
['the', 'car', 'of']
        Mean: 0.304, Var: 0.101
['the', 'car', 'of', 'the']
        Mean: 0.179, Var: 0.035
['the', 'car', 'of', 'the', 'driver']
        Mean: 0.03, Var: 0.001
['the', 'car', 'of', 'the', 'driver', 'with']
        Mean: 0.027, Var: 0.001
['the', 'car', 'of', 'the', 'driver', 'with', 'the']
        Mean: 0.011, Var: 0.0
['the', 'car', 'of', 'the', 'driver', 'with', 'the', 'mustache']
```

```
        Mean: 0.008, Var: 0.0
Ambiguous attachment:
['the']
        Mean: 0.5, Var: 0.25
['the', 'son']
        Mean: 0.335, Var: 0.112
['the', 'son', 'of']
        Mean: 0.304, Var: 0.101
['the', 'son', 'of', 'the']
        Mean: 0.179, Var: 0.035
['the', 'son', 'of', 'the', 'driver']
        Mean: 0.03, Var: 0.001
['the', 'son', 'of', 'the', 'driver', 'with']
        Mean: 0.027, Var: 0.001
['the', 'son', 'of', 'the', 'driver', 'with', 'the']
        Mean: 0.011, Var: 0.0
['the', 'son', 'of', 'the', 'driver', 'with', 'the', 'mustache']
        Mean: 0.005, Var: 0.0
```