There is more than one way to search working memory

Pedro Alcocer

Draft: July 25, 2011

1 Introduction

What kind of algorithm is used to search sentence representations in working memory during online sentence processing? The alternatives we consider are (i) a serial algorithm that searches the sentence representation by traversing its graph structure and (ii) a parallel algorithm that queries the entire representation at once. The experiments and computational model described here provides evidence that contributes to this debate. We exploit a grammatical constraint in Brazilian Portuguese that requires a specific long-distance structural relation between a null subject and its antecedent. Whether an algorithm is serial or parallel has consequences for the kinds of relations that the algorithm is sensitive to. Long-distance relations—c-command, in particular—may be problematic for a parallel algorithm, but not a serial algorithm. In the case of locating a null subject’s antecedent, an algorithm’s sensitivity to long-distance structure should determine whether it considers a structurally ineligible subject as a possible antecedent. A serial algorithm should not consider the structurally ineligible subject, while a parallel algorithm, which is potentially blind to long-distance relations, should show signs of having considered the ineligible subject during the retrieval.

In three experiments and a computational model, we present evidence from null subject licensing and agreement processing that the human sentence processor has available to it both kinds of algorithm and is selective in its choice of algorithm. We also present evidence that a computational model which assumes a parallel algorithm cannot capture the fallibility profiles seen in humans even
when given potentially unreasonable help in the form of relational cues.

2 Search algorithms

Completing the grammatical dependencies that emerge during parsing requires the rapid and repeated retrieval of material from a sentence’s developing representation. For retrieval to take place, the location of the to-be-retrieved elements of the representation in memory must be identified, which requires search. There are two dominant hypotheses for how the kind of working memory that is used during sentence processing might be searched. Search may proceed serially using an algorithm that explicitly navigates a sentence’s phrase structure graph in order to locate a queried element. Or, search may proceed in parallel, through a cue-based algorithm that has access to all elements of the representation at once in the context of a content-addressable memory.

A recent series of studies suggests that the sentence processor builds, manipulates, and stores linguistic representations in a content-addressable workspace (McElree, 2000; McElree et al., 2003; Lewis and Vasishth, 2005; Lewis et al., 2006). Throughout this paper we proceed with the assumption that, regardless of the algorithm used to search it, the memory architecture underlying retrieval is content-addressable, though we make no strong commitments to its instantiation. We also make the assumption that the atomic unit of memory retrieval is the attribute-value pair, or the “cue”. Cues are bundled together into a “chunk”\(^1\), which, after Lewis and Vasishth (2005), we conceive of as an associative array: a set of valued attributes.

![Figure 1: A chunk with several cues. The left column lists the attributes this chunk contains. The right column lists the values associated with those attributes.](image)

\(^1\)We will tend to use the word “chunk” discussing encoding matters and “node” when discussing graph-theoretic matters, but they are interchangeable.
The standard conception of memory from a computational perspective is random-access memory (RAM). An intuitive way to conceive of how data is stored in RAM is as items, or data, placed in a series of consecutively numbered boxes, or memory addresses. Each memory address contains a piece of data, say, a string of letters. If I want to retrieve the string “elephant”, I have to know which address contains it. If I know it is in address 7, I can look there and retrieve the string. Otherwise, I have to search. If the data in the addresses is not ordered, I have to look in each address, one by one, until I encounter “elephant”. This is search in RAM.

Search in content-addressable memory (CAM) proceeds differently. With CAM, I can look at all the data at once. To search, I issue the query “elephant” and the memory returns all the addresses that contain that string. The parallel search algorithm we are considering is more complex than this, but is still based on the principle of being able to examine all stored data at once. A more detailed description of the algorithm is provided in Section 5.1, which describes ACT-R modeling.

These assumptions—especially the content-addressability assumption—are required for parallel search, but not serial search. Parallel search requires content-addressability. Serial search does not. Nothing about CAM, however, precludes the emulation of serial search within such an architecture. Serial search can be run in CAM straightforwardly given that chunks can be made to be homologous to nodes in a graph and traversed in the same way: by following edges between nodes. These edges can be explicitly encoded in a chunk. For example, a chunk can point to the chunk that represents its parent node in a directed graph through a cue that consists of a “parent” attribute that is valued with the unique ID value of the chunk’s parent. Search is a series of parallel searches for unique ID values. We are not committed to this particular implementation of the serial algorithm, but offer it as an existence proof for serial search in CAM.

Figures 2a and 2b illustrate how these search algorithms might proceed for locating the antecedent of a reflexive pronoun in a sentence like (1). In this example, the parser detects a reflexive at himself and, using the grammar to construct a search query, launches a search for its antecedent. In this case, the antecedent must be a local NP that c-commands the reflexive pronoun and has singular
and masculine φ-features. These criteria are expressed as explicit cues in Figure 3.

(1) The man who hit Bill identified himself.

Figure 2: Search algorithms.

Figure 3: A search query for the antecedent of the reflexive anaphor himself.

The serial algorithm seen in Figure 2a follows the c-command path up the tree, starting at the reflexive pronoun, checking at every node for the relevant cues and getting directions to next eligible node. The parallel algorithm seen in Figure 2b queries all the in-memory chunks at once and returns those that match the search query. In this example, the identity of the returned element is indicated by a 1.
Several important properties of each algorithm become apparent. The serial search algorithm is able to navigate the representation more efficiently if it imposes an order to the cues. It can ignore whole, irrelevant swathes of the representation by applying the c-command and locality constraints first. For instance, the NP Bill is a good candidate for himself’s antecedent in terms of category and \( \phi \)-features, but due to its position in the relative clause within the subject NP, that node is not even touched upon by the serial algorithm. On the other hand, the parallel algorithm cannot help but consider Bill as a potential antecedent. In fact, given the number of its cues that match with the search query, Bill is the most likely antecedent after the true antecedent, the man who hit Bill.

As we will discuss later, being able to ignore irrelevant parts of the representation means that the information which is eventually retrieved is more likely to be grammatically faithful. This property of serial search, grammatical faithfulness, supplies us with an empirical toehold into discerning which kind of algorithm is being used which we will take advantage of later.

(2) **C-command** (Reinhart, 1976): A c-commands B iff

i. A does not dominate B and B does not dominate A;

ii. the first branching node that dominates A dominates B.

Next, the facility with which the algorithms capture a long-distance relation (i.e., a relation between a node and another node more than one degree away) like c-command, (2), is another point of divergence between them. The algorithms make different demands of the representations they search over. Under the serial algorithm, c-command is conceived of as an implicit relationship between two nodes in a graph. This is in keeping with the serial algorithm’s respect for the sentence representation as a hierarchical object. To the parallel algorithm, the sentence representation is flat. Under the parallel algorithm, whether c-command is present between a node and another node is a fact about those nodes and requires explicitly encoding of that information as a cue. Also, because it is not necessarily known ahead of time what kind of dependencies a chunk will enter into, this means that a chunk could potentially contain a record of its c-command status with every other
There seems to be a tradeoff between computational complexity and representational complexity. Complex algorithms require simple representations, while simple algorithms require complex representations. Assuming representational complexity is optimized for each algorithm (i.e., the representation is just sufficiently rich enough for the algorithm to achieve its goal), the serial algorithm is more computationally complex than the parallel algorithm. In order to compute c-command between two nodes, it must go through at least \( n \) iterations, where \( n \) is the number of nodes between the c-commander and the c-commanded. On the other hand, the parallel algorithm has exactly one iteration. The catch, however, is that in order for the parallel algorithm to accurately capture c-command in just one step, the representations it is searching over have to be extremely rich. Each chunk would have to explicitly encode its c-command status with respect to every other chunk in memory. Comparatively, the representations required for the accurate capture of c-command under the serial algorithm are meager. Chunks only need to point to their immediate parents and children.

It may seem like the choice of which complexity to favor is an aesthetic question, but this is not the case. First, it could be argued that the cognitive system may better be able to handle an algorithm whose computational complexity grows linearly than one whose memory requirements grow quadratically, as the parallel algorithm’s do. More importantly, however, in positing extremely rich representations we are merely shunting over the work of computing c-command relations to the system that is in charge of encoding the representations in the first place. Absent heuristics, with every new element that enters the representation, the “chunk-maker” must compute c-command relations between the chunk and every other chunk in the representation and potentially update old

\[ \text{chunk in the representation.}^2 \]

\[^2\text{It is not clear, given the assumptions we currently make about the data structures that underlie cues and chunks how a chunk’s c-command relation with respect to another chunk could be encoded using attribute-value pairs. In such a relationship there are two variables that need to be encoded: the identity of the potential c-commanded chunk and whether that chunk is indeed c-commanded. One way to encode both variables, would be to store one of them in the attribute, e.g. } [\text{c-command} \rightarrow \text{DP2: +, c-command} \rightarrow \text{DP3: −, c-command} \rightarrow \text{DP4: +}]. \text{ Perhaps erroneously, we have been considering the attributes that can be included in a chunk to be a closed set. Another possibility is to store the IDs of c-commanded chunks in a list under a single attribute, e.g., } [\text{c-command: } [\text{DP2, DP4}]]. \text{ This, of course, raises questions about memory limitations. In any case, absent a theory of cues, it is difficult to draw limits on what is representable by cues and how. This may be a premature consideration, however.} \]
chunks with new information. We are again left with the problem of navigating memory in order to compute relations between elements, only this time, in the homunculus that is the encoding system. The problem enters an infinite regress. For this reason, from a conceptual standpoint, it is preferable to favor computational complexity, and therefore, a serial search algorithm.

It should be made clear that a parallel search algorithm should not have such a hard time with all structural relations, only those whose members can be arbitrarily far apart from each other hierarchically. A more local relation, like clausematehood, could conceivably be encoded using cues. It would require that the parser be able to use a counter to keep track of the current clause depth. With the start of every new clause the counter would be incremented and at the end of every clause, it would be decremented. Each chunk would be assigned a clause index according to the value of the clause counter at the time of their creation. In order to locate chunks that were clausemates of another chunk, a cue would be added to the search query that specified the appropriate clause index. Using this strategy would allow a parallel algorithm to use structural relations into order to faithfully complete dependencies without having to resort to a graph-based incremental search as the serial algorithm would. This fact becomes relevant when we try to use grammatical faithfulness as an indicator that search occurred serially. Faithfulness serves that function only under specific circumstance which will be discussed in the following sections.

3 Dependencies

The grammar establishes constraints on which elements can enter into dependency relations. The role that these constraints play in online processing seems to vary as a function of the type of dependency that is being completed. Experimental evidence indicates that there is a great deal of variability in how accurately grammatical constraints are implemented online. One dimension that seems to account for some of it is the directionality of dependencies.

Grammatical dependencies have two elements, which we will call the left- and right-hand elements,
based on their linear position in a sentence string. Generally, one of these elements alerts the parser that a dependency exists and the second element is the one that needs to be located in order to complete the dependency. There are cases when the left-hand element is the one that signals that a dependency is present, as in (3). These are considered forwards dependencies because the parser must look forwards in time in order to locate the second element of the dependency. In contrast, in a backwards dependency, like (4), the right-hand element is the signal. The parser must look backwards in time (i.e., into memory) in order to locate the second element and complete the dependency. It should be made clear that we describe dependencies in this way because it is useful to and not because we take these to notions to have serious theoretical standing.

(3) While he ate, John watched TV. FORWARDS DEPENDENCY

(4) John watched TV while he ate. BACKWARDS DEPENDENCY

The completion of a dependency is said to be unfaithful when processing the dependency results in actions that do not respect constraints in the grammar. Such actions include temporarily considering an ineligible element for retrieval, positing a gap in an ineligible position, or fully retrieving an incorrect element from memory. This is in contrast to a faithful dependency, where the grammar is fully respected during processing. Examples of both of these types of dependency will be discussed in the coming sections.

There is evidence that the directionality of a dependency—that is, whether the signaling element comes first or second—is correlated with the faithfulness of the dependency. Phillips et al. (2009) present evidence that forwards dependencies are more faithful that backwards dependencies. They claim that prospective, prospective search is a more robust algorithm for locating the other end of a dependency than retrospective search in memory.

Our strategy in this paper is to use faithfulness data to tease out the kind of search algorithm that is in use by the sentence processor. In the next subsections we present an example of a robust
forwards dependency, cataphora, and several examples of backwards dependencies with mixed faithfulness profiles. We conclude the section with an elaboration of why the dependencies that have been used thusfar to study search algorithms via faithfulness have not been the ideal ones to use.

3.1 Cataphora, a forwards dependency

One forwards-looking dependency that seems to be fairly robust against interference and to consistently respect grammatical constraints is backwards anaphora, or cataphora. In the more standard forwards anaphora, antecedents precede pronouns as in (5). In cataphora, the arrangement is reversed.

Unlike anaphora, cataphoric dependencies are forward-looking because, in these cases, the element that signals the dependency (i.e., the pronoun) precedes the element that completes it (the antecedent), so when the signaling element is identified, a second, completing element is expected, as in (6). A cross-linguistically well-attested constraint on backwards anaphora is Binding Principle C (Chomsky, 1981), which prevents an R-expression and a c-commanding pronoun from coreferring, as in (7). If the parser respects Principle C online, then it should not consider R-expressions in the c-command path of a cataphoric pronoun as possible antecedents for that pronoun.

(5) John, watched TV while he, ate.

(6) While he, ate, John, watched TV.

(7) *He, said that John, saw the movie

In a self-paced reading study, Kazanina et al. (2007) presented evidence that Principle C is respected in online processing. In sentences like (8), the authors crossed two factors: (i) whether the pronoun and the antecedent matched in gender and (ii) whether there was a c-command relation between the pronoun and its potential antecedent. The first factor was introduced to exploit the Gender Mismatch Effect, which causes a slowdown in reading time if two elements that should agree in gender do not agree. The second factor varied whether the Principle C constraint could come into
effect by having the pronoun c-command the R-expression [(8-c), (8-d)] or not [(8-a), (8-b)].

The prediction was that if Principle C is respected online, then reading times at Jessica and Russell should not be significantly different in (8-c) and (8-d), where Principle C prohibits coreference, but should be different in (8-a) and (8-b), where coreference is permissible. In (8-c) and (8-d), Jessica and Russell are ineligible antecedents for the pronoun because the pronoun c-commands them. Therefore, they should not be considered when an antecedent is being sought out as the second element in the dependency. On the other hand, Principle C is not in effect in (8-a) and (8-b) because there is no c-command relation between the pronoun and the antecedent. Therefore, the R-expressions are eligible antecedents of she. Being eligible, the parser considers them when searching out an antecedent and makes them liable to the Gender Mismatch Effect. In this case, Russell should have slower reading times than Jessica. The results of the study hold true to this prediction, suggesting that Principle C is reflected in online processing.

(8)  
   a. While she was taking classes full-time, Jessica was working two jobs to pay the bills.  
   b. While she was taking classes full-time, Russell was working two jobs to pay the bills.  
   c. She was taking classes full-time while Jessica was working two jobs to pay the bills.  
   d. She was taking classes full-time while Russell was working two jobs to pay the bills.

These results show that the parser can respect grammatical constraints online and serve as a foundation piece for ensuing studies that rely on the parser's ability to respect grammatical constraints online for their argumentation.

3.2 Backward dependencies

It is tempting to assume that there is a one-to-one mapping between a dependency's faithfulness and the kind of search algorithm used to complete that dependency. The argument is that if the search algorithm that locates elements in memory respects hierarchical representations, then dependencies that rely on structural relations for their licensing should be faithfully retrieved; structure should lead
the search algorithm to the element it is trying to locate. On the other hand, if the search algorithm is blind to structural information, then dependencies that rely on structural relations for their licensing should be unfaithfully retrieved; parallel search ignores some kinds of structure and, unlike serial search, is able to retrieve elements that are in grammatically ineligible positions as long as cue match is sufficiently strong. Tempting though this is to assume, it is simply not the case, as has been noted in previous sections. A parallel algorithm can use cues to encode some structural relations and, therefore, can, in some cases, faithfully execute a structurally-constrained search. When faithful retrieval of this sort takes place, we have no way of telling the two search algorithms apart.

The following sections discuss studies that examine different dependencies for their faithfulness.

3.2.1 Agreement

Studies looking at subject-verb agreement over a distance suggest that the parser does not always take advantage of structural cues (Pearlmutter et al., 1999; Wagers et al., 2009; Alcocer and Phillips, 2009). Though both are ungrammatical due to subject-verb agreement mismatch, readers are faster to read (9-b) than (9-a) and more likely to judge it as acceptable. Accounts of the exact mechanism vary, but according to one of the dominant accounts, in (9-b), when performing a backwards search to find the verb's subject, the parser incorrectly retrieves the plural feature on runners and illicitly completes the dependency (Wagers et al., 2009, cf. BOCK CITATION).

(9) a. The runner who the driver see every morning always wave say say hi.
   b. The runners who the driver see every morning always wave say say hi.

This could be taken as evidence for a parallel search algorithm. Structure should be enough to guide a search from a verb to its subject, yet the fact that agreement is unfaithful suggests that structure is not used as a guide.
3.2.2 Reflexives

One case of faithful backwards dependency completion is reported in Sturt (2003). Binding Principle A (Chomsky, 1981) requires that anaphors like *himself* have a c-commanding antecedent in the same clause. In (10), upon reaching the anaphor, the parser must make a retrieval into memory to find an antecedent. If Principle A is respected online, then the structurally ineligible DP *Jennifer/Jonathan* should not be considered in the search for an antecedent (there should be no reading time slow downs at the reflexive). In (10), the subject *surgeon* is stereotypically biased towards masculine referents. There is evidence that readers are subject to this bias during online comprehension. This bias might lead readers to consider *Jennifer* as the antecedent to *herself* despite the fact that it is structurally ineligible. However, Sturt's results show that readers did not consider the matching, but structurally ineligible intervener, providing evidence that Principle A is respected online and that retrieval is structure-guided.

(10)  
   a. The surgeon that treated Jonathan pricked himself with a needle  
   b. The surgeon that treated Jennifer pricked herself with a needle  
   c. The surgeon that treated Jonathan pricked herself with a needle  
   d. The surgeon that treated Jennifer pricked himself with a needle

This could be taken as evidence for structure-guided search. In the case of reflexives, it seems that the search algorithm uses structure to locate the appropriate antecedent. This is evidenced by the fact that the intervening potential antecedent is not considered at all during retrieval.

3.2.3 Negative polarity items

Negative polarity items (like *any* or *ever*) are licensed by a c-commanding negative element\(^3\), as in (11-a). NPIs are not licensed when not c-commanded by a negative element, as in (11-b) (negative

\(^3\)This is a simplification. NPIs are licensed by elements that create downward entailing contexts. The c-command requirement is a side effect of how semantic interpretation licenses such entailments.)
element absent) or (11-c) (negative element not c-commanding).

(11)  
   a. No professor will ever say that.
   b. *A professor will ever say that.
   c. *A professor that no student likes will ever say that.

   Studies across languages and paradigms have shown, however, that NPI licensing can be interfered upon by a linearly intervening non-c-commanding element (Drenhaus et al., 2005, et seq.). For instance, in a speeded grammaticality judgment study, Xiang et al. (2006) found that participants were 15–30% more likely to judge a sentence like (12-c) as acceptable than (12-b).

(12)  
   a. No bills that the senator voted for will ever become law.
   b. *The bills that the senators voted for will ever become law.
   c. *The bills that no senators voted for will ever become law.

   The c-command relation is structural and should not be fallible if memory is searched using structure. Vasishth et al. (2008) have taken the fallibility of NPI licensing as evidence for cue-based retrieval, though it is not clear what the search cues in this case would be exactly. Xiang et al. (2009) suggest that erroneous semantic/pragmatic inferences may be to blame, not retrieval.

3.3 Shortcomings

Unfortunately, none of these cases are water-tight, knock-down evidence for either kind of search algorithm. The culprit has been locality. Agreement requires that the two elements of the dependency be in the same clause. Reflexives share this feature. It is possible that elements within the same clause are specially encoded and that a parallel algorithm takes advantage of this encoding. Or, the faithfulness seen with reflexives might be due to bona fide serial search. It is not possible to discern these two. These studies, therefore, are not conclusive evidence for either kind of search algorithm.

   Further, it is not obvious that NPI licensing requires retrieval. Licensing of this kind of dependency
may rely on some semantic or pragmatic computation which makes no reference to retrieval. If this is the case, then evidence from NPI licensing is not useful evidence for determining the nature of the search algorithm.

The crucial study that has not been conducted is one which features a dependency where the elements are not clausemates (unlike agreement and reflexives), where the licensing conditions are clearly syntactic (unlike NPI licensing), and where retrieval is clearly required (also unlike NPI licensing). The present study therefore should serve as a key piece of evidence in the debate about how memory is searched.

4 The present study

4.1 Background

In partial null subject (NS) languages, null subjects are licensed under more restricted conditions than full NS languages. Holmberg et al. (2009), examining several partial NS languages, reduce the conditions under which null subjects are licensed in these languages to two: (i) when the subject is a generic pronoun corresponding to English “one” and (ii) when the subject is controlled by the immediately higher overt subject.

In (13) we see the consequences of condition (ii) on a partial NS language, Brazilian Portuguese, and a full NS language, Spanish. In (13-a), the pronoun ele corefers with the subject João. In a non-null subject language, like English, the pronoun is obligatorily overt. In full and partial NS languages, the pronoun is obligatorily null. In (13-b), however, full and partial NS languages differ. The pronoun in these example could be optionally overt or null in full NS languages, but crucially, would be obligatorily overt in partial NS languages. To restate condition (ii): the NS must be c-commanded by an immediately higher overt subject for the NS to be bound to that subject; an unbound NS is illicit.

(13) a. (i) João, disse que *ele/pro, tinha comprado uma casa (BP)
(ii) Juan dijo que *el/pro, había comprado una casa (Sp.)

b. (i) João disse que ele/*pro, tinha comprado uma casa (BP)

(ii) Juan, dijo que el/pro, había comprado una casa (Sp.)
    J. said that he/pro had bought a house.
    “J. said that he bought a house.”

C-command is necessary but not sufficient to license NS coreference. It really must be the immediately higher overt c-commanding subject. In (14-a), we know that the NS is intended to corefer with the non-immediate c-commanding subject João because the DP rico has masculine gender marking. This is ungrammatical. On the other hand, in (14-b), we know that the NS is intended to corefer with the immediately higher overt subject (i.e., the first c-commanding subject that is not pro), Maria. This is grammatical. Crucially for our goals, there is no requirement that the null subject and its antecedent subject be clausemates, as (14-b) illustrates.

(14) a. *João acha que Maria pensa que é rico
    J. thinks that M. thinks that is rich-masc

    b. João acha que Maria pensa que é rica
    J. thinks that M. thinks that is rich-fem
    “J thinks that M. thinks that he/she is rich”

In the present study, we exploit this second condition in order to test whether backwards search uses an algorithm that is sensitive to a long-distance relation, like c-command. With such an algorithm, locating the appropriate c-commander is straightforward and should proceed faithfully. Importantly, however, parallel search is blind to the c-command relation, making it impossible to selectively and faithfully retrieve the appropriate c-commander.

With BP null subjects, we have a dependency that lacks the confounds present in previous study, allowing us to get a clean look at the kind of algorithm that is used during search. In order to establish a c-command relation between a null subject and an overt c-commanding DP, there must be some dependence on structure in memory search. Given that the null subject and the antecedent subject are not clausemates, we can be certain that no special encoding that is triggered by clausalmatehood...
is in play. Given that the antecedent subject precedes the null subject, which is the signaling element of the dependency, making this a backwards dependency, we can be sure that a retrieval must take place. This puts us in an excellent position to determine what kind of algorithm is used in search. The only cue to the identity of the antecedent subject available to the search algorithm is structure. If we find evidence that the parser considers the intervening, structurally ineligible subject, this is evidence for parallel search. On the other hand, if we find evidence that no such a consideration is made, then this is evidence for serial search.

4.2  Experiment 1: Acceptability judgments on null subject licensing

Experiment 1 aims to test two major assumptions of this study. First, that the generalization about the constraints on BP null subjects made explicit by Holmberg et al. (2009) hold true for the tested population in Rio de Janeiro. Second, that this population is sensitive to number agreement. For practical reasons it is important to test these assumptions given that the claim that null subjects in BP are as restricted as they are is not widely known among speakers the language and given that Holmberg et al. claim that European Portuguese (EP) is a full null subject language which does not have the same null subject restrictions as BP.

In (15), the null subject has two c-commanding subjects, but only the one in the clause immediately higher to it (Maria) should be a legal antecedent to the NS. Therefore, an adjective agreeing with the NS should agree with the immediately-higher c-commanding subject, and not any others. If participants are as sensitive to the NS constraint as Holmberg et al. argue, then we should see much higher acceptance rates for (15-b) than (15-a). Though we do not test sentences with this exact template, we do test sentences that contain the same constraint.

(15)   a. *João, acha que Maria pensa que pro, é rico
          João thinks that Maria thinks that pro is rich.masc
          “J. thinks that M. thinks that he is rich.”
b. João acha que Maria pensa que *pro* é *rica*
   João thinks that Maria thinks that *pro* is rich.
   “I thinks that M. thinks that she is rich.”

Naro and Scherre (2000), among others, have also noted that there is some variability in number morphology production in BP speakers. In particular, some dialects of BP mark number agreement only on the pronoun, but not on the verb (16-a); only on the determiner, but not on the noun (16-b) or the adjective (16-c) or the verb (16-d); on the determiner and the noun, but not the adjective (16-e); on the pronoun and the noun, but not the determiner (16-f). It is possible that these production phenomena extend to comprehension, leading to a decrease in number sensitivity compared to speakers of Standard BP.

(16)  
   a. *ele* ganha *demais*  
       *3PL* win, *3SG* too-much  
       “They win too much”

   b. *as* *codorna*  
      *DET.PL* quail, *SG*  
      “the quails”

   c. *as* *porta* *aberta*  
      *DET.PL* door, *SG* open, *SG*  
      “the open doors”

   d. *as* *coisa* *tá* *cara*  
      *DET.PL* thing, *SG* be, *3SG* expensive, *SG*  
      “Things are expensive”

   e. *essas* *estradas* *nova*  
      *DET.PL* roads, *PL* new, *SG*  
      “Those new roads”

   f. *do* *meus* *pais*  
      *of DET.SG* my, *PL* parents, *PL*  
      “of my parents”

In this experiment, we employ three conditions: (1) a grammatical condition where the critical
verb agrees with an eligible subject but not an ineligible subject, (2) an ungrammatical condition where the critical verb agrees with neither the eligible or ineligible subject, and (3) and ungrammatical condition where the critical verb agrees with an ineligible subject but not an eligible subject. Eligibility refers to the subject's position in the sentence; either in the matrix clause (eligible to be an antecedent of the NS) or a relative clause (ineligible). The details of these are elaborated upon in a following section.

The only acceptability profile that would indicate that participants are sensitive to the null subject licensing constraint and to agreement relations is one where the grammatical condition is judged to be significantly more acceptable than the two ungrammatical conditions. An acceptability profile where one or both of the ungrammatical conditions patterns with the grammatical condition would indicate a lack of sensitivity to either the null subject constraint or to agreement.

4.2.1 Participants

Participants were 41 native speakers of Brazilian Portuguese from the Universidade Federal do Rio de Janeiro community with no history of language disorders. All participants provided informed consent. Participants were compensated R$10 per hour for their participation. Some participants also participated in Experiment 3 — which did not share materials with the present experiment — during the same session. No participant took part in more than one experimental session.

4.2.2 Materials

The materials consisted of 18 sets of sentences in a 3-condition design with eligible DP match (levels: match/mismatch) and ineligible DP match (levels: match/mismatch) as factors. The combination eligible match, ineligible match was not used because we considered it redundant with the eligible match, ineligible mismatch condition. This yielded three conditions, which, for the sake of clarity and brevity, will be called ELIGIBLE MATCH (eligible match, ineligible mismatch), NO MATCH (eligible mismatch, ineligible mismatch), and INELIGIBLE MATCH (eligible mismatch, ineligible match) from
now on. These three conditions were designed to assess the degree to which the constraints on null subjects posited by Holmberg et al. (2009) are present in the grammar of our test population and to establish that this population is sensitive to agreement. Examples sentences from each of the conditions can be seen in (17) which can all be glossed as (18). In all conditions, we hold the retrieval cue — the plural verb estavam, in this example — constant and manipulate whether previous DPs in eligible and ineligible positions match in number features with it. The ineligible DP is in an object relative clause and is so called because, given the null subject constraint, it should not be eligible as an antecedent to a downstream null subject in a position not c-commanded by this DP. In the eligible match condition (17-a), the eligible DP matches in features with the retrieval cue, in the no match condition (17-b), nothing matches with the cues, and in the ineligible match condition (17-c), only the ineligible DP matches with the cues.

Every experimental item always followed the same form: DP, complementizer que, DP, embedded verb. This was followed by a PP ranging in length from 3 to 5 words designed to separate the embedded and matrix verbs. This was then followed by the matrix verb, the complementizer que, the critical verb, and a PP continuation.

(17) a. Os turistas que [o ladrão enganou na rua deserta] perceberam que pro estavam numa área ruim da cidade. 

b. O turista que [o ladrão enganou na rua deserta] percebeu que pro estavam numa área ruim da cidade.

c. O turista que [os ladrões enganaram na rua deserta] percebeu que pro estavam numa área ruim da cidade.

(18) The tourist(s) that the thief{f|ves} fooled on the deserted street noticed that they were in a bad part of the city.
<table>
<thead>
<tr>
<th>Condition</th>
<th>Eligible Subject DP</th>
<th>Ineligible Subject DP</th>
<th>Verb</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELIGIBLE MATCH</td>
<td>PL</td>
<td>SG</td>
<td>PL</td>
</tr>
<tr>
<td>NO MATCH</td>
<td>SG</td>
<td>SG</td>
<td>PL</td>
</tr>
<tr>
<td>INELIGIBLE MATCH</td>
<td>SG</td>
<td>PL</td>
<td>PL</td>
</tr>
</tbody>
</table>

Table 1: Number features of the critical items in Experiment 1’s design.

The 18 sets of 3 conditions were distributed across 3 lists in a Latin Square design, and combined with 36 grammatical fillers. This resulted in 22% of the items being ungrammatical.

4.2.3 Procedures

Participants were given a paper questionnaire and asked to rate sentences from 1 (very bad) to 7 (very good) by circling a number under each sentence.

4.2.4 Analysis

An exact Wilcoxon Mann-Whitney rank sum test (Hollander and Wolfe, 1999) was run on each pair of conditions. Given that rating data is not normally distributed, this test, which makes no assumptions about the distribution of the data, is preferable to one that assumes a normal distribution.

One participant was excluded for rating all items with a 7 rating.

4.2.5 Results

Figure 4 summarizes the results of Experiment 1. Responses from the ELIGIBLE MATCH condition skewed significantly towards the “very good” end of the rating continuum compared to the NO MATCH and INELIGIBLE MATCH conditions. The ELIGIBLE MATCH condition was significantly different from the NO MATCH ($Z = 8.86, p << 0.001$) and the INELIGIBLE MATCH conditions ($Z = 7.30, p << 0.001$). The NO MATCH and INELIGIBLE MATCH conditions did not differ significantly from each other ($Z = -1.58, p = 0.11$). A significant difference remains when the NO MATCH and INELIGIBLE MATCH conditions are grouped together and compared with the ELIGIBLE MATCH condition ($Z = 9.44, p << 0.001$).
Figure 4: Histogram of ratings from Experiment 1 by condition. Mean ratings: Good (5.9), Bad (4.5), Interferer (4.9).

4.2.6 Discussion

The results suggest that readers were able to discern a difference between the Eligible Match condition and, together, the NO MATCH and INELIGIBLE MATCH conditions, rating the Eligible Match condition significantly better than the other two. We still, however, see more right-skewedness (i.e., skewedness towards “good”) than expected in the ungrammatical conditions.4

Such a pattern of results would only be possible if this population of participants were both sensitive to the posited null subject constraint and sensitive to number agreement generally. If

Diogo Almeida (p.c.) notes that if a DP is extremely discourse prominent and proximal to the null subject, the NS can be licensed. A sentence like (i) is licit, but one like (ii) is not. It is possible that since participants had unlimited time to render a judgment, that they generously posited such a discourse prominent antecedent when they were judging the sentences, making some of them better than they otherwise would have been.

(i) Remember John,? I think pro, will win the race.

(ii) *Remember Mary, whose son John, lives on the house on the corner? I think pro, will win the race.

---

4Diogo Almeida (p.c.) notes that if a DP is extremely discourse prominent and proximal to the null subject, the NS can be licensed. A sentence like (i) is licit, but one like (ii) is not. It is possible that since participants had unlimited time to render a judgment, that they generously posited such a discourse prominent antecedent when they were judging the sentences, making some of them better than they otherwise would have been.
participants were not sensitive to the null subject constraint, we might expect the **ineligible match** condition to be rated significantly higher than the **no match** condition, which is not the case. If participants were no sensitive to number agreement, we might expect the **no match** and **ineligible match** conditions to pattern with the **eligible match** condition, which is not the case. With these results in hand, we have good evidence that the null subject constraint can be used to test retrieval questions.

### 4.3 Experiment 2: Processing null subjects

The aim of this experiment is to assess how Brazilian Portuguese speakers respect the null subject constraint online. We are interested in determining whether they ever consider the ineligible DP when attempting to retrieve the null subject’s antecedent from memory. The way the materials have been designed, the **eligible match** condition should always be grammatical and the **no match** condition should always be ungrammatical. The **ineligible match** condition, however, should be ungrammatical, but is potentially, depending the architecture of the memory system underlying the retrieval, liable to intrusion by the ineligible subject embedded in the relative clause in the matrix subject DP. Upon encountering the null subject, the parser must look for a licensor. If the parser uses structure in its search, then it should not consider the structurally ineligible intervener DP and the sentence should be judged as ungrammatical. However, if some other search algorithm is employed — one that ignores structure to some degree — then the structurally ineligible DP may be considered as the null subject licensor and the sentence may pass as grammatical.

If search is entirely guided by structure, then the condition in which a number-agreeing, but structurally ineligible DP intervenes between the verb and the subject head DP (**ineligible match**) should be just as slow as a condition where neither of the subject DPs before the verb agrees with it. This should return a pattern of reading times like those in (19).

\[(19) \quad \text{eligible match} < \text{ineligible match} = \text{no match}\]
If cues other than structure are predominantly used in search, then a structurally ineligible, but number-agreeing subject DP might be temporarily considered as the antecedent of the null subject. In this case, we would expect that, on average, reading times at the critical word in the ineligibility match condition would be somewhere between the average reading times at the same word in the eligible match condition (fastest) and the no match condition (slowest).

(20)  ELIGIBLE MATCH < INELIGIBLE MATCH < NO MATCH

4.3.1 Participants

Participants were 77 native speakers of Brazilian Portuguese from the Universidade Federal do Rio de Janeiro community with no history of language disorders. All participants provided informed consent. Participants were compensated R$10 (≈ 5.50 USD) per hour for their participation. No participant also took part in Experiment 1.

4.3.2 Materials

The materials were the same as those from Experiment 1. Region definitions are reported in Table 2.

<table>
<thead>
<tr>
<th>Condition / Region</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eligible Match</td>
<td>Os turistas que o ladrão enganou na rua deserta perceberam que estavam numa área ruim da cidade.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No Match</td>
<td>O turista que o ladrão enganou na rua deserta percebeu que estavam numa área ruim da cidade.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ineligible Match</td>
<td>O turista que os ladrões enganaram na rua deserta percebeu que estavam numa área ruim da cidade.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Regions.

4.3.3 Procedures

Sentences were presented on a laptop PC using the Linger software (Doug Rohde, MIT) in a self-paced word-by-word moving window paradigm (¿). Each trial began with a screen presenting a sentence in which the words were masked by dashes. Each time the participant pressed the space bar, a word was revealed and the previous word re-masked. A yes/no comprehension question appeared in its entirety after each sentence. The “F” key was used for “yes” and the “J” key was used for “no”. On-screen feedback was provided for incorrect answers. Participants were instructed to
read at a natural pace and answer the questions as accurately as possible. Order of presentation was randomized for each participant. Five practice items were presented at the start of the experimental session.

4.3.4 Analysis

Exclusions One participant was excluded for having a mean accuracy less than 80% on comprehension questions. Two participants were excluded for having mean reading times greater than 3 standard deviations from the overall reading time mean. Three participants were excluded for having a too low a slope on a measure that regressed reading time against word length, which suggests that they were pressing keys rhythmically and not as they read. Three additional participants were excluded due to equipment error or abnormalities during the testing session. In total nine participants were excluded from the original, leaving 68 participants for further analysis.

A z-score was computed for each region in each trial based on the mean and standard deviation of each region across all trials. Regions with a z-score greater than 2 or less than –2 were excluded. Inaccurate trials were excluded.

Statistical analyses In order to minimize the effect of a trial’s position in the list and the length of the word associated with a trial on that trial’s reading time, a linear mixed-effect model was fit to each region’s reading times using word length and log list position as fixed effects and participants as a random effect. The residuals of this model were used for further analyses.

Because we employed an unbalanced design (one eligible condition vs. two ineligible conditions), factors were recoded using contrast coding into the factors “match within ineligible” and “eligibility”. With this coding, the match factor were compared only within the ineligible conditions and the ineligible conditions were grouped and compared against the eligible condition.
### Table 3: Contrast coding used in analysis of Experiment 2.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Match</th>
<th>Eligibility</th>
<th>Contrast 1</th>
<th>Contrast 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELIGIBLE MATCH</td>
<td>+</td>
<td>+</td>
<td>0</td>
<td>+2</td>
</tr>
<tr>
<td>NO MATCH</td>
<td>−</td>
<td>−</td>
<td>−1</td>
<td>−1</td>
</tr>
<tr>
<td>INELIGIBLE MATCH</td>
<td>+</td>
<td>−</td>
<td>−1</td>
<td>−1</td>
</tr>
</tbody>
</table>

A mixed-effects model was fit for each region with residual reading times as the response variable, the two contrast-coded factors as fixed effects, and participants and items as random effects.

#### 4.3.5 Results

A table of \( p \)-values resulting from the mixed-effects models fit for each region can be seen in Table 4. Figure 5 summarizes the reading time results of Experiment 2.

**Regions 1–3.** No significant effects were found in these regions.

**Region 4.** Contrast 1 was significant in this region. We saw a significant increase in reading times for the INELIGIBLE MATCH condition. This may be because this was the only condition with a plural verb in this region. We have seen previous evidence of slowdowns attributed to plurals (CITATION).

**Regions 5–6.** No significant effects were found in these regions.

**Region 7.** Contrast 1 was significant in this region. We saw a significant increase in reading times for the INELIGIBLE MATCH condition. This effect was short-lived; the separation between ineligible conditions was not present in the following region.

**Regions 8–10, critical regions.** Region 8 was the critical region where we would first expect to see an effect. There was a numerical trend toward a separation of the ineligible conditions (NO MATCH and INELIGIBLE MATCH) from the eligible condition (ELIGIBLE MATCH). This trend reached significance in Region 9, where Contrast 2, which compared the eligibility factor, showed a significant difference. The numerical trend continued in Region 10, but the significant effect did not.
Figure 5: Experiment 2 results.
<table>
<thead>
<tr>
<th>Region</th>
<th>Contrast</th>
<th>MCMC Mean</th>
<th>Lower 95% CI</th>
<th>Upper 95% CI</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>-3.6705</td>
<td>-15.257</td>
<td>8.335</td>
<td>0.5410</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.4674</td>
<td>-6.292</td>
<td>7.225</td>
<td>0.8916</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-1.3780</td>
<td>-14.621</td>
<td>11.958</td>
<td>0.8322</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-0.6951</td>
<td>-8.057</td>
<td>7.107</td>
<td>0.8612</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1.019</td>
<td>-10.44</td>
<td>12.286</td>
<td>0.8556</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-1.036</td>
<td>-7.14</td>
<td>5.669</td>
<td>0.7494</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>-37.636</td>
<td>-70.97</td>
<td>-5.826</td>
<td>0.0206</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-16.078</td>
<td>-34.25</td>
<td>2.729</td>
<td>0.0858</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>4.675</td>
<td>-7.787</td>
<td>16.95</td>
<td>0.4496</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3.292</td>
<td>-3.873</td>
<td>10.28</td>
<td>0.3588</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2.9190</td>
<td>-21.88</td>
<td>26.36</td>
<td>0.8154</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-0.9743</td>
<td>-14.90</td>
<td>11.96</td>
<td>0.8886</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>-17.5389</td>
<td>-35.43</td>
<td>-0.8582</td>
<td>0.0464</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-5.5622</td>
<td>-15.46</td>
<td>4.1608</td>
<td>0.2634</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0.1162</td>
<td>-18.48</td>
<td>17.533</td>
<td>0.9914</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-7.5967</td>
<td>-18.13</td>
<td>2.346</td>
<td>0.1398</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>4.3506</td>
<td>-12.39</td>
<td>21.182</td>
<td>0.6060</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-14.8704</td>
<td>-24.09</td>
<td>-4.817</td>
<td>0.0022</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>-0.8232</td>
<td>-11.477</td>
<td>9.5880</td>
<td>0.8680</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-5.3482</td>
<td>-11.092</td>
<td>0.7935</td>
<td>0.0768</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>2.096</td>
<td>-12.18</td>
<td>16.021</td>
<td>0.7684</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-6.465</td>
<td>-14.44</td>
<td>1.326</td>
<td>0.1120</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>-6.398</td>
<td>-23.09</td>
<td>9.7398</td>
<td>0.4428</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-8.545</td>
<td>-18.08</td>
<td>0.2101</td>
<td>0.0698</td>
</tr>
</tbody>
</table>

Table 4: Experiment 2 results. The double line demarcates the critical region.

4.3.6 Discussion

According to our hypotheses, a pattern of reading times where eligible match was read faster than either ineligible match or no match (which should not have been significantly different) would indicate that a search algorithm that was sensitive to long-distance structural relations was in use (see (19)). We saw such a pattern in the results of Experiment 2, specifically in Region 9, the region following the critical verb.

The significant difference in Region 9 suggests that participants were not illicitly considering the DP located in an ineligible relative clause when searching for the antecedent of the null subject. We
interpret this finding as evidence that the parser must have available to it a serial search algorithm that navigates sentential structure.

The verb at Region 8 was the first evidence that participants received that a null-subject existed. In order to correctly compute the agreement at the verb, the parser must locate its subject. Given the nature of the null-subject constraint, we argue that the only way for the parser to correctly retrieve that appropriate c-commanding DP is by taking advantage of the structural information provided by the sentence. We argue that it is not possible to encode a long-distance relation like c-command the same way a morphological feature like number or case might be encoded.

4.4 Experiment 3: Agreement

4.4.1 Participants

Participants were 46 native speakers of Brazilian Portuguese from the Universidade Federal do Rio de Janeiro community with no history of language disorders. All participants provided informed consent. Participants were compensated R$10 (∼ 5.50 USD) per hour for their participation. No participant took part in more than one experimental session.

4.4.2 Materials

The materials consisted of 24 sets of sentences of 4 conditions in a $2 \times 2$ design with grammaticality (levels: grammatical, ungrammatical) and attractor (levels: attractor present, attractor absent) as factors. A template of the materials can be seen in (21). $\text{DP}_1$, the true subject, was always singular, while $\text{DP}_2$, the non-local intervener, and $\text{V}_2$, the embedded subject, varied in number independently. $\text{V}_1$ always agreed grammatically with $\text{DP}_1$. A sample item can be seen in (22). Also included were 36 fillers, half of which were grammatical. The errors in the ungrammatical fillers were not related to agreement. The overall grammatical to ungrammatical ratio was 1:1.

$$\text{DP}_1 \left[ \text{RC} \text{DP}_2 \text{ } \text{V}_2 \text{PP} \right] \text{V}_1 \text{PP}$$
(22)  a. O turista que o ladrão enganou na rua deserta estava numa área ruim da cidade.
    
    det.sg tourist.sg that det.sg thief.sg fooled.sg in-the street deserted was.sg in-a area bad of-the city

b. Os turistas que o ladrão enganou na rua deserta estavam numa área ruim da cidade.
    
    det.pl tourist.pl that det.sg thief.sg fooled.sg in-the street deserted was.pl in-a area bad of-the city

c. *O turista que o ladrão enganaram na rua deserta estava numa área ruim da cidade.
    
    det.sg tourist.sg that det.sg thief.sg fooled.pl in-the street deserted was.sg in-a area bad of-the city

d. *Os turistas que o ladrão enganaram na rua deserta estavam numa área ruim da cidade.
    
    det.pl tourist.pl that det.sg thief.sg fooled.pl in-the street deserted was.sg in-a area bad of-the city

<table>
<thead>
<tr>
<th>Grammatical Attractor</th>
<th>Intervener DP</th>
<th>True subject DP</th>
<th>Verb</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>-</td>
<td>SG</td>
<td>[SG  SG]</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>PL</td>
<td>[SG  SG]</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>SG</td>
<td>[SG  PL]</td>
</tr>
<tr>
<td>-</td>
<td>+</td>
<td>PL</td>
<td>[SG  PL]</td>
</tr>
</tbody>
</table>

Table 5: Number features of the critical items in Experiment 3’s design.

4.4.3 Procedures

Sentences were presented one word at a time in the center of the screen with rapid serial visual presentation (RSVP) reading (?) at a rate of 200 ms per word. At the end of each sentence, a response screen appeared for two seconds, during which participants made a yes/no response by pressing a button. Participants were instructed to read the sentences carefully and to judge whether they sounded like an acceptable sentence that a speaker of Brazilian Portuguese might say. Participants were instructed to make their responses as quickly as possible following the end of the sentence. If two seconds elapsed without a response, participants were given feedback that their response was
too slow.

4.4.4 Analysis

Mean accuracy on the filler items before exclusions was 66%. After excluding 13 participants for having less than 60% accuracy on the filler items, the mean accuracy was 72%. After these exclusions, 33 participants remained. All analyses were performed on the data with exclusions.

A logistic mixed-effects model was fit with response (“yes” or “no”) as the response variable, number nested under each grammaticality level (“grammatical”, “ungrammatical”) as the fixed effects, and participant and item as random effects. Defining the fixed effects term of the model this way allows us to examine the two levels of grammaticality separately, which would not be possible with the typical “main effects plus interaction” model definition.

4.4.5 Results

Figure 6 shows the percent of “yes” (i.e., “acceptable”) responses for the four conditions in Experiment 3. Within the grammatical conditions, there is no significant difference between the conditions with a singular or plural intervenor DP ($p = 0.25$). Within the ungrammatical conditions, however, there is a significant difference between the condition with a singular intervenor DP and the condition with a plural intervenor DP ($p << 0.001$), with the plural condition having a significantly greater percentage of “yes” responses. Unsurprisingly, an overall main effect of grammaticality ($p << 0.001$) was also found.
4.4.6 Discussion

Experiment 3 replicates the results reported by Wagers et al. (2009). Sentences with ungrammatical subject-verb agreement with plural intervenors are significantly more likely to be misperceived as acceptable than similar sentences with singular intervenors, an illusion of acceptability. We also replicate the grammaticality asymmetry that Wagers et al. report: the above effect is only present in the ungrammatical items. There is no illusion of unacceptability.

The results suggest that Brazilian Portuguese speakers are not able to faithfully retrieve an appropriate subject in the face of a plural intervenor.

4.5 Discussion

We can make three empirical points from the previous experiments.
1. Brazilian Portuguese speakers from the Rio de Janeiro region are sensitive to agreement in comprehension, and, crucially, sensitive to the null-subject constraint described by Holmberg et al. (2009).

2. These same speakers are able to ignore grammatically illicit positions when searching for a null subject's antecedent.

3. But, these same speaker are not able to ignore grammatically illicit positions when searching for a verb's subject in order to check agreement.

Taken together, points 2 and 3 present a puzzle: under some circumstances readers are able to ignore grammatically irrelevant material, and under others they are not. These two cases are an especially useful pair because in both of them the same kind of material is being searched for, subject DPs, and the same cues are triggering the search, verbs with morphological number features. Because these two cases are so similar, we must then ask: what conditions are necessary for grammatically faithful (or unfaithful) retrieval? We are not able to answer this question with the present data, but we are able to speculate more precisely. The present results seem to make the case that the sentence processor is neither perfectly faithful or unfaithful to the grammar. We are able to rule out accounts that predict one kind of faithfulness profile all of the time. These results suggest that the sentence processor can be selective about what search algorithm to employ. The question now becomes, how does the parser decide what kind of search algorithm to use?

It may a matter of the interpretational consequences of retrieving the wrong item during search. Locating the antecedent of a null subject is a kind of anaphora resolution. The semantic content of the item that ends up being retrieved will be reused in the clause that contains the anaphoric null subject. For this reason, misretrieving a null subject's antecedent could cause serious interpretational problems. Checking agreement, on the other hand, may have fewer interpretational consequences. The goal of checking agreement is only to make sure that the relevant \( \phi \)-features of the two items in question match. No semantic material is moved around and interpretation is unaffected. Therefore,
getting agreement wrong is relatively innocuous compared to getting a null subject’s antecedent wrong. The sentence processor may go through the extra effort that serial search requires in order to avoid interpretational boondoggles.

Another possibility is that the deciding factor is locality. The sentence processor may employ a serial algorithm to search for non-local dependencies, which are difficult to encode for in the sentence representation, and a parallel algorithm to search for local ones, which are easier to encode for.

A problem with these accounts is that the material that triggers search in both the null subject case and the agreement case is the same: an inflected verb. If the triggers are matched, it is not clear how the sentence processor should decide which kind of search to use. However, there is one way in which these two cases are not exactly matched. The null subject case has at least one additional processing step. After parsing the verb, the parser must notice that it lacks an overt subject and posit a null subject. It then proceeds to search memory for the overt antecedent of the null subject. This additional step may provide an additional clue to a pre-search algorithm that makes a decision regarding what kind of search algorithm to employ about how important it is to get retrieval right or about the expected position of the antecedent.

5  Computational modeling

In this section we use computational modeling evidence to make two claims:

1. The effects we report in Experiment 2 cannot be explained merely by the relative activation in memory of the relevant DPs at the moment when the reader reaches the critical verb.

2. The same model using the same set of parameters cannot capture the faithfulness profiles seen in humans when processing agreement and null subject dependencies, even when provided with relational cues.
The first claim makes reference to the fact that it is logically possible that, in materials used in Experiment 2, the effect that we report is being driven by an increase in relative activation of the eligible DP compared to the ineligible DP at the moment when the critical verb is reached and a search is launched for the null subject's antecedent. In these materials, there is a verb (the matrix verb) before the critical verb that reactivates the matrix subject. In (24-a), for example, the verb *perceberam*, though not the critical verb, activates the eligible subject DP *os turistas*, but does not necessarily activate the ineligible subject DP *o ladrão*. Thus, by the time the reader reaches the critical verb *estavam*, there is a potential bias in activation towards the eligible subject. If this activation bias correlates with the empirical results (i.e., in the cases where the correct DP is retrieved, that DP happens to have a higher activation at the critical verb), then it is not possible to conclude that sensitivity to structure is what is driving the effect — it may be a mere activation effect. It should be noted that, by design, this verb agrees with the matrix subject in all conditions.

The second claim addresses the more interesting question of whether it is possible for a parallel search algorithm to capture the pattern of faithfulness that Experiment 2 suggests humans are capable of *and* capture the pattern on unfaithfulness seen in Experiment 3. The results of Experiment 2 suggest that humans are able to complete the BP null subject dependency faithfully — they are not led astray by interfering, structurally ineligible and grammatically illicit material. Is there an arrangement under which a system using only parallel search could replicate this pattern of faithfulness? Conversely, Experiment 3 suggests that human participants are prone to misretrieve a subject NP when faced with an intervening NP. In this case is seems that grammatical faithfulness is abandoned.

To address these questions, we modeled sentences like those in (23), in the case of agreement, and (24), in the case of null subjects, using the Adaptive Character of Thought - Rational (ACT-R) framework (Anderson and Lebiere, 1998; Anderson, 2005), which has been used before for this kind of modeling (Lewis and Vasishth, 2005; Vasishth et al., 2008; Patil et al., 2011; Dillon, 2011, *inter alia*). We take the following approach: First, we model retrieval using only non-relational cues. These include category, φ-features, and structural position (e.g., spec TP). What cues like these have in
common is that they should be straightforwardly deliverable by the parser. That is, they can be read off the morphology, or, in the case of the structural position, do not require a computation that involves more than adjacent nodes in the phrase structure. Next, we model retrieval using a cue, “embeddedness”, that requires an assumption that the parser is passing along information about the chunk’s position in the sentence structure relative to other, potentially non-adjacent chunks. The embeddedness cue maps onto whether the element that the chunk represents is in a matrix position or not (i.e., embedded or not).

(23) \[ \text{DP}_1 [RC \; \text{DP}_2 \; V_2 \; PP \; ] \; V_1 \; PP \]

  a. O turista que o ladrão \textit{enganou} na rua deserta estava numa
\textit{DET.SG} tourist.SG that \textit{DET.SG} thief.SG \textit{fooled.SG} in-the street deserted \textit{was.SG} in-a
\textit{area ruim da cidade.}
\textit{area bad of-the city}

  b. Os turistas que o ladrão \textit{enganou} na rua deserta estavam numa
\textit{DET.PL} tourist.PL that \textit{DET.SG} thief.SG \textit{fooled.SG} in-the street deserted \textit{was.PL} in-a
\textit{area ruim da cidade.}
\textit{area bad of-the city}

  c. \textit{*O turista que o ladrão \textit{enganaram} na rua deserta estava numa
\textit{DET.SG} tourist.SG that \textit{DET.SG} thief.SG \textit{fooled.PL} in-the street deserted \textit{was.SG} in-a
\textit{area ruim da cidade.}
\textit{area bad of-the city}}

  d. \textit{*Os turistas que o ladrão \textit{enganaram} na rua deserta estavam
\textit{DET.PL} tourist.PL that \textit{DET.SG} thief.SG \textit{fooled.PL} in-the street deserted \textit{was.SG}
\textit{numa área ruim da cidade.}
\textit{in-a area bad of-the city}}

(24) \[ \text{DP}_1 [RC \; \text{DP}_2 \; V_2 \; PP \; ] \; V_1 [RC \; pro \; V \; PP] \]

  a. Os turistas que [o ladrão \textit{enganou} na rua deserta] perceberam que \textit{pro estavam numa
\textit{area ruim da cidade.}}

  b. \textit{*O turista que [o ladrão \textit{enganou} na rua deserta] percebeu que \textit{pro estavam numa área
ruim da cidade.}}

  c. \textit{*O turista que [os ladrões \textit{enganaram} na rua deserta] percebeu que \textit{pro estavam numa
area ruim da cidade.}}
The representations that ACT-R computes over are called chunks. A chunk is a bundle of features that enters into relations with other chunks. Chunks can have varying levels of activation. The overall activation of chunk is computed according to Equation 1, which is a sum of the chunk’s baseline activation ($B_i$), the additional activation it receives as a function of cue-match with the search query ($\sum_j W_j S_{ji}$), mismatch penalties ($\sum_k PM_{ki}$), and noise ($\xi$). This additional activation is a function of $W_j$, the weights of the cues in the search query, and $S_{ji}$, the strengths of association between cues $j$ and chunk $i$. The latter value is arrived through Equation 2. Where $S$ is the maximum associative strength and $fan_j$ is the “fan” parameter, which is a function of the total number of items associated with the cue $j$. So, as a cue is more often used in a sentence representation, the associative strength that comes from that cue is reduced. Mismatch penalties are a function of $P$, a typically negative parameter that determines how much activation is lost due to cue mismatch, and $M_{ji}$, a logical value that is 1 if cue $j$ does not match with chunk $i$ and 0 otherwise. Finally, $\xi$ is a noise parameter that models the stochasticity of real processing. It follows a logarithmic distribution with a mean of 0 and a variance that is a function of the noise parameter $s$.

$$A_i = B_i + \sum_j W_j S_{ji} + \sum_k PM_{ji} + \xi$$

$$S_{ji} = S - \log(fan_j)$$

$$\sigma^2 = \frac{\pi}{3} s^2$$

The baseline activation is arrived at by Equation 4, where $t_j$ is the time that has elapsed since
the \( j \)th retrieval of the chunk \( i \), and \( d \) is the decay parameter, which by convention is always set to 0.5 (Anderson, 2005; Lewis and Vasishth, 2005). This parameter causes the activation of a chunk to decay logarithmically as time goes on and the chunk is not reactivated.

\[
B_i = \log \left( \sum_{j=1}^{n} t_j^{-d} \right)
\]

(4)

A chunk's overall activation also determines its activation “latency”, \( T_i \), described by Equation 5. Here \( F \) is a constant scaling factor that can be manipulated to better fit model latencies to empirical latencies, and after Lewis and Vasishth (2005) is set to 0.14.

\[
T_i = Fe^{-A_i}
\]

(5)

Modeling retrieval in ACT-R involves providing the framework with two matrices for each condition being tested: one that represents the chunks in the sentence and one that describes the retrieval schedule.

5.2 Materials

The sentence representations and retrieval schedules used in modeling are listed in Appendix A.

5.3 Results

5.3.1 Model error and acceptability

Model error refers to the proportion of 5000 trials in which the model retrieves an interferer item instead of the correct item. Model error is plotted as a box and whiskers plot that represents the probability distribution over percent error for a particular condition. This representation allows us to summarize the results of many models that have been run over a set of parameters.
Translating from model error to acceptability  The mean model error returned by the ACT-R model should not be confused with acceptability as judged by humans (particularly the sort of acceptability elicited in a Speeded Acceptability Judgment Task, as in Experiment 3). Unlike humans, the model is aware of which DP is the correct one because this fact is explicitly encoded in the model definition. When the model retrieves the wrong DP, this is counted as a mis-retrieval and the mean model error is increased. When judging acceptability, on the other hand, humans can only go by the features of the retrieved item to make their decision. A mis-retrieved item might still be judged as acceptable if its features match the retrieval cues.

We can translate from model error to acceptability by taking into account the model error (i.e., the proportion of the time that the wrong item is retrieved) and whether the features which the retrieved item carries match the retrieval features. If they do, we assume that a human would judge this item as acceptable. If they do not, we assume it would be judged unacceptable. We are aware that feature match is not the only factor that decides acceptability. Indeed, we seldom see perfectly good or perfectly bad acceptability profiles. Ratings often come just short of that. Nonetheless, we are comfortable making the assumption that feature match is a major factor in the acceptability computation, at least when it comes to agreement.

In Table 6, we give an example of translating to acceptability in the case of agreement without a structural cue. In condition -gram, -int, neither the correct DP or the intervening DP (both singular) matches in number features with the retrieval cue (plural). This means that no matter what DP is retrieved, the DP will never match in number with the retrieval cue and therefore yields 0% acceptability. In the -gram, +int condition, the correct DP (singular) does not match in number features with the retrieval cue (plural), but the intervening DP (plural) does. The model retrieves the intervening DP 82.8% of the time, meaning that the retrieved DP matches in features with the retrieval cue 82.8% of the time, and therefore yields an acceptability of 82.8%. In the +gram, -int condition, the correct and intervening DPs both match in features (singular) with the retrieval cue (singular), therefore, no matter what DP is retrieved, it will always match the retrieval cue, yielding
100% acceptability. Finally, in the -gram, +int condition, the correct DP (singular) matches in features with the retrieval cue (singular), but the intervening DP does not (plural). The model retrieves the intervening DP 8.8% of the time. The rest of time, it retrieves the correct DP, which therefore yields a 91.2% acceptability.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Correct DP number</th>
<th>Intervening DP number</th>
<th>Retrieval cue</th>
<th>Model error: % of time retrieved wrong DP</th>
<th>Acceptability: % of time retrieved DP matches retrieval cue</th>
</tr>
</thead>
<tbody>
<tr>
<td>-gram, -int</td>
<td>SG</td>
<td>SG</td>
<td>PL</td>
<td>39.9%</td>
<td>0.0%</td>
</tr>
<tr>
<td>-gram, +int</td>
<td>SG</td>
<td>PL</td>
<td>PL</td>
<td>82.8%</td>
<td>82.8%</td>
</tr>
<tr>
<td>+gram, -int</td>
<td>SG</td>
<td>SG</td>
<td>SG</td>
<td>40.3%</td>
<td>100.0%</td>
</tr>
<tr>
<td>+gram, +int</td>
<td>SG</td>
<td>PL</td>
<td>SG</td>
<td>8.8%</td>
<td>91.2%</td>
</tr>
</tbody>
</table>

Table 6: Example of translating from model error to acceptability.

5.3.2 Agreement models

Model error profiles for the agreement models are reported in Figure 7 and the acceptability profiles derived from the model error are reported in Figure 8. Here, we refer to the + and – levels of the interference factor from Experiment 3 as “plural” and “singular” in order for the relationship between the condition and the features used in retrieval to be more transparent.

In the conditions that do not use a structural cue during retrieval, within the grammatical conditions, the singular condition is more prone to errors (40.8%) than the plural condition (17.8%). This is expected because in the grammatical, plural condition, only the true subject matches perfectly with the retrieval cues. In the singular condition, both the eligible and ineligible subjects match in number with the retrieval cues. This difference in error rates within the grammatical condition is not evident in humans because in the singular condition, the content of what they end up retrieving matches the retrieval cues no matter what, masking the error.

Within the ungrammatical conditions, the singular condition is less prone to errors (40.7%) than the prone condition (69.5%). This follows from the fact that in the plural intervener condition the
retrieval cue is plural and the eligible subject is singular, making the ineligible DP the more attractive choice. In the singular condition, the retrieval cue is again plural and the eligible subject is singular. This leaves the search algorithm with no good option with respect to feature match. The choice of what to retrieves comes down to the other terms in the activation equation. Here, the algorithm effectively flips a coin. If there is a bias towards the eligible subject it is due to its linear proximity to the verb.

In the conditions that make use of the structural embeddedness cue, the error profile is the same, though overall error is reduced because the algorithm is now getting some help from the extra cue. The structural cue does not make a difference in terms of the ordering of conditions.

![Figure 7: Agreement model error.](image)
Figure 8: Derived acceptability for the agreement model.
Moving on to the derived acceptability data reported in Figure 8, it seems that the would-be acceptability profile derived from model error rates matches the empirical acceptability profile seen in humans processing agreement as reported in Figure 6 and repeated for convenience in Figure 9. These modeling results suggest that the model is successfully able to capture the lack of grammatical faithfulness seen in humans processing agreement.

5.3.3 Null subject models

Model error profiles for the null subject models are reported in Figure 10 and the acceptability profiles derived from the model error are reported in Figure 11. Here, we refer to the ELIGIBLE MATCH condition as the grammatical condition, the INELIGIBLE MATCH condition as match possible condition, and the NO MATCH condition as match not possible condition in order to make the distinction between INELIGIBLE MATCH and NO MATCH more salient. These are both ungrammatical
conditions whose only difference is that in one, there is a DP in the representation with the appropriate \( \phi \)-features, and in the other, there is not.

The overall model error pattern seems to be that the grammatical condition is the least error prone (with structural cue: 21.8%, without: 37.9%), the match not possible condition is intermediate (37.4%, 52.1%), and match possible condition is the most error prone (46.5%, 69.7%). Looking at the derived acceptability profiles, the model would predict humans processing null subjects to find the grammatical condition very acceptable (79.2%, 62.1%), unsurprisingly. Also not surprising is that the model would predict humans to find the match not possible condition not acceptable at all (0%, 0%). In the match possible condition, the model would predict high levels of acceptance for the match possible condition (46.5%, 69.6%). In fact, in the conditions without the structural cue, the model predicts that humans should prefer the ungrammatical match possible condition over the grammatical condition (69.6% vs. 62.1%). Even with the structural cue, there are high levels of acceptance in the match possible condition.

We do not have empirical acceptability data to compare these predictions against, but we do have offline judgment data as reported in Experiment 1. In that data, the grammatical condition is preferred over both ungrammatical conditions. Under the modeling predictions, we might expect an intermediate acceptability profile for the match possible condition — not as acceptable as the grammatical condition, but more acceptable than the no match possible condition — but the judgment data show no significant difference between these ungrammatical conditions.

These results indicate that the ACT-R model does not capture the same faithfulness profile that is seen in humans processing null subjects. Further, these results indicate that the reading time profile seen in at least the region after the critical region in Experiment 2
Figure 10: Null subject model error.

Figure 11: Derived acceptability for the null subject model.
5.4 Conclusions

The two claims we sought to make in this section were that (i) the effects we reported in Experiment 2 could not be reduced to an activation account and that (ii) the same model using the same set of parameters could not capture the faithfulness profiles seen in humans when processing agreement and null subject dependencies, even when provided with relational cues.

6 Conclusions

In this paper we presented three experiments: two on the processing of the null subject dependency in Brazilian Portuguese and one on the processing of agreement dependencies in BP. We also presented a computational model of each of these dependencies.

Experiment 1 provided offline judgment evidence that BP speakers in the Rio de Janeiro region are sensitive to the null subject constraint described by Holmberg et al. (2009). Experiment 2 established that BP speakers were sensitive to this constraint online, suggesting that a search algorithm that was capable of capturing long distance dependencies was in use: a serial search algorithm. Experiment 3 showed that BP speakers can be insensitive to some grammatical constraints, namely agreement. A computational model of agreement processing in the ACT-R framework predicted such results, which is in line with the hypothesis that participants were using a parallel search algorithm to search for the verb’s subject. The same computational model, when presented with a task that required the location of a null subject’s antecedent, failed to predict empirical results (provided by Experiments 1 and 2) which suggested that participants were sensitive to grammatical constraints that work over long distances and that they had this sensitivity because they were using a serial search algorithm.

Together, these results suggest that the human sentences processor has available to it both kinds of search algorithms: serial and parallel. They also suggest that the processor is selective about its use of serial search, reserving it for when interpretation could be negatively affected by an incorrect retrieval or for when a dependency required long distance sensitivity for faithful completion.
A Materials used in modeling

This appendix contains the sentence representations and retrieval schedules for all the conditions that were modeled in Section 5. Each set of sentence representations and retrieval schedules was run with and without the embeddedness cue.

The various rows in the sentence representation matrix correspond to name, a unique identifier for chunk; created, the timepoint in which the chunk comes into existence; cat, the chunk’s grammatical category; number, the chunk’s number feature; and role, the chunk’s position in the sentence structure. In the implementation of ACT-R we are using, only the features name, created, and cat are required. The rest can be defined at will. The special value nil indicates that the cue for that particular feature is empty or missing from the chunk. The feature name is not used in the activation computation.

In the retrieval schedule matrix, the various rows correspond to the features that are being retrieved over and moment, the timepoint at which the retrieval is initiated.

A.1 Null subject modeling

<table>
<thead>
<tr>
<th>name</th>
<th>DP1</th>
<th>DP2</th>
<th>VP1</th>
<th>PP1</th>
<th>DP3</th>
<th>VP2</th>
<th>VP3</th>
<th>PP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>created</td>
<td>0</td>
<td>300</td>
<td>600</td>
<td>900</td>
<td>1200</td>
<td>1500</td>
<td>1800</td>
<td>2100</td>
</tr>
<tr>
<td>cat</td>
<td>DP</td>
<td>DP</td>
<td>VP</td>
<td>PP</td>
<td>DP</td>
<td>VP</td>
<td>VP</td>
<td>PP</td>
</tr>
<tr>
<td>number</td>
<td>PL</td>
<td>SG</td>
<td>SG</td>
<td>nil</td>
<td>SG</td>
<td>PL</td>
<td>PL</td>
<td>nil</td>
</tr>
<tr>
<td>role</td>
<td>specT</td>
<td>specT</td>
<td>headVP</td>
<td>headPP</td>
<td>compP</td>
<td>headVP</td>
<td>headVP</td>
<td>headPP</td>
</tr>
<tr>
<td>(embed)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 7: ELIGIBILE MATCH condition sentence representation.

<table>
<thead>
<tr>
<th>moment</th>
<th>1650</th>
<th>1950</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat</td>
<td>DP</td>
<td>DP</td>
</tr>
<tr>
<td>number</td>
<td>PL</td>
<td>PL</td>
</tr>
<tr>
<td>role</td>
<td>specT</td>
<td>specT</td>
</tr>
<tr>
<td>(embed)</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 8: ELIGIBILE MATCH condition retrieval schedule.
Table 9: NO MATCH condition sentence representation.

Table 10: NO MATCH condition retrieval schedule.

Table 11: INELIGIBLE MATCH condition sentence representation.

Table 12: INELIGIBLE MATCH condition retrieval schedule.
A.2 Agreement modeling

<table>
<thead>
<tr>
<th>name</th>
<th>DP</th>
<th>VP</th>
<th>PP</th>
<th>DP</th>
<th>VP</th>
<th>PP</th>
<th>DP</th>
<th>VP</th>
<th>PP</th>
<th>DP</th>
</tr>
</thead>
<tbody>
<tr>
<td>created</td>
<td>300</td>
<td>600</td>
<td>900</td>
<td>1200</td>
<td>1500</td>
<td>1800</td>
<td>2100</td>
<td>2400</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cat</td>
<td>DP</td>
<td>DP</td>
<td>VP</td>
<td>PP</td>
<td>DP</td>
<td>VP</td>
<td>PP</td>
<td>DP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>number</td>
<td>PL</td>
<td>SG</td>
<td>SG</td>
<td>nil</td>
<td>SG</td>
<td>PL</td>
<td>nil</td>
<td>SG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>role</td>
<td>specT</td>
<td>specT</td>
<td>headVP</td>
<td>headPP</td>
<td>compPP</td>
<td>headVP</td>
<td>headPP</td>
<td>compPP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(embed)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 13: +gram, +int condition sentence representation.

| moment  | 1050 | 1950 |
| cat     | DP   | DP   |
| number  | SG   | PL   |
| role    | specT | specT |
| (embed) | 1     | 0    |

Table 14: +gram, +int condition retrieval schedule.

<table>
<thead>
<tr>
<th>name</th>
<th>DP</th>
<th>VP</th>
<th>PP</th>
<th>DP</th>
<th>VP</th>
<th>PP</th>
<th>DP</th>
<th>VP</th>
<th>PP</th>
<th>DP</th>
</tr>
</thead>
<tbody>
<tr>
<td>created</td>
<td>300</td>
<td>600</td>
<td>900</td>
<td>1200</td>
<td>1500</td>
<td>1800</td>
<td>2100</td>
<td>2400</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cat</td>
<td>DP</td>
<td>DP</td>
<td>VP</td>
<td>PP</td>
<td>DP</td>
<td>VP</td>
<td>PP</td>
<td>DP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>number</td>
<td>SG</td>
<td>SG</td>
<td>SG</td>
<td>nil</td>
<td>SG</td>
<td>SG</td>
<td>nil</td>
<td>SG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>role</td>
<td>specT</td>
<td>specT</td>
<td>headVP</td>
<td>headPP</td>
<td>compPP</td>
<td>headVP</td>
<td>headPP</td>
<td>compPP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(embed)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 15: +gram, -int condition sentence representation.

| moment  | 1050 | 1950 |
| cat     | DP   | DP   |
| number  | SG   | SG   |
| role    | specT | specT |
| (embed) | 1     | 0    |

Table 16: +gram, -int condition retrieval schedule.
<table>
<thead>
<tr>
<th>name</th>
<th>DP1</th>
<th>DP2</th>
<th>VP1</th>
<th>PP1</th>
<th>DP3</th>
<th>VP2</th>
<th>PP2</th>
<th>DP4</th>
</tr>
</thead>
<tbody>
<tr>
<td>created</td>
<td>300</td>
<td>600</td>
<td>900</td>
<td>1200</td>
<td>1500</td>
<td>1800</td>
<td>2100</td>
<td>2400</td>
</tr>
<tr>
<td>cat</td>
<td>DP</td>
<td>DP</td>
<td>VP</td>
<td>PP</td>
<td>DP</td>
<td>VP</td>
<td>PP</td>
<td>DP</td>
</tr>
<tr>
<td>number</td>
<td>PL</td>
<td>SG</td>
<td>PL</td>
<td>nil</td>
<td>SG</td>
<td>PL</td>
<td>nil</td>
<td>SG</td>
</tr>
<tr>
<td>role</td>
<td>specT</td>
<td>specT</td>
<td>headVP</td>
<td>headPP</td>
<td>compPP</td>
<td>headVP</td>
<td>headPP</td>
<td>compPP</td>
</tr>
<tr>
<td>(embed)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 17: -gram, +INT condition sentence representation.

| moment | 1050 | 1950 |
| cat | DP | DP |
| number | PL | PL |
| role | specT | specT |
| (embed) | 1 | 0 |

Table 18: -gram, +INT condition retrieval schedule.

<table>
<thead>
<tr>
<th>name</th>
<th>DP1</th>
<th>DP2</th>
<th>VP1</th>
<th>PP1</th>
<th>DP3</th>
<th>VP2</th>
<th>PP2</th>
<th>DP4</th>
</tr>
</thead>
<tbody>
<tr>
<td>created</td>
<td>300</td>
<td>600</td>
<td>900</td>
<td>1200</td>
<td>1500</td>
<td>1800</td>
<td>2100</td>
<td>2400</td>
</tr>
<tr>
<td>cat</td>
<td>DP</td>
<td>DP</td>
<td>VP</td>
<td>PP</td>
<td>DP</td>
<td>VP</td>
<td>PP</td>
<td>DP</td>
</tr>
<tr>
<td>number</td>
<td>SG</td>
<td>SG</td>
<td>PL</td>
<td>nil</td>
<td>SG</td>
<td>SG</td>
<td>nil</td>
<td>SG</td>
</tr>
<tr>
<td>role</td>
<td>specT</td>
<td>specT</td>
<td>headVP</td>
<td>headPP</td>
<td>compPP</td>
<td>headVP</td>
<td>headPP</td>
<td>compPP</td>
</tr>
<tr>
<td>(embed)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 19: -gram, -INT condition sentence representation.

| moment | 1050 | 1950 |
| cat | DP | DP |
| number | PL | SG |
| role | specT | specT |
| (embed) | 1 | 0 |

Table 20: -gram, -INT condition retrieval schedule.
References


