The role of memory in superiority violation gradience

Marisa Ferrara Boston
Cornell University
Ithaca, NY, USA
mfb74@cornell.edu

Abstract
This paper examines how grammatical and memory constraints explain gradience in superiority violation acceptability. A computational model encoding both categories of constraints is compared to experimental evidence. By formalizing memory capacity as beam-search in the parser, the model predicts gradience evident in human data. To predict attachment behavior, the parser must be sensitive to the types of nominal intervenors that occur between a wh-filler and its head. The results suggest memory is more informative for modeling violation gradience patterns than grammatical constraints.

1 Introduction
Sentences that include two wh-words, as in Example (1), are often considered difficult by English speakers.

(1) *Diego asked what1 who2 read?

This superiority effect holds when a second wh-word, who in this example, acts as a barrier to attachment of the first wh-word and its verb (Chomsky, 1973).

The difficulty is ameliorated when the wh-words are switched to which-N, or which-Noun, form as in Examples (2) and (3) (Karttunen, 1977; Pesetsky, 1987). This is confirmed by experimental evidence (Arnon et al., To appear; Hofmeister, 2007).

(2) ?Diego asked which book who read?
(3) ?Diego asked what which girl read?

Memory is often implicated as the source of this gradience, though it is unclear which aspects of memory best model experimental results. This computational model encodes grammatical and memory-based constraints proposed in the literature to account for the phenomenon. The results demonstrate that as memory resources are increased, the parser can model the human pattern if it is sensitive to the types of nominal intervenors. This supports memory-based accounts of superiority violation (SUV) gradience.

2 Explanations for SUV gradience
This section details grammatical and reductionist explanations for SUV gradience, motivating the encoding of various constraints in the computational model.

2.1 Grammatical explanations
Grammatical accounts of gradience rely on intrinsic discourse differences between phrases that allow for SUVs and those that do not. In this work, which-N phrases are examples of the former, and so-called bare wh-phrases (including who and what) the latter. Rizzi (1990) incorporates ideas from Pesetsky’s D-Linking, or discourse-linking, hypothesis (1987) into a grammatical account of SUV gradience, Relativized Minimality. He argues that referential phrases like which-N refer to a pre-established set in the discourse and are not subject to the same constraints on attachment as non-referential phrases, like what. Which book delimits a set of possible discourse entities, books, and is more restrictive than what, which could instead delimit sets of books, cats, or abstract entities. The Relativized Minimality hypothesis accounts for SUV gradience on the basis of this categorical separation on wh-phrases in the discourse.

Both bare phrases and which-N phrases could have the appropriate discourse conditions to allow for superiority violations, and vice versa. However, to relate the theory’s predictions to the experiment modeled here, I use a categorical split between which-N and bare wh-phrases.
2.2 Reductionist explanations

Many grammatical accounts, particularly those that are grounded in cognitive factors, incorporate some element of processing or memory in their explanations (Phillips, Submitted). Reductionist accounts are different; their proponents do not believe that superiority requires a grammatical explanation. Rather, SUVs that appear ungrammatical, such as Example (1), are the result of severe processing difficulty alone.

These accounts attribute processing difficulty to memory: severe memory resource limitations account for ungrammatical sentences in SUVs, and increased memory resources allow for more acceptable sentences. This is the central idea behind Hofmeister’s Memory Facilitation Hypothesis (2007):

Memory Facilitation Hypothesis
Linguistic elements that encode more information (lexical, semantic, syntactic, etc.) facilitate their own subsequent retrieval from memory (Hofmeister, 2007, p.4)\(^2\).

This memory explanation is central to activation-based memory hypotheses previously proposed in the psycholinguistic literature, such as CCREADER (Just and Carpenter, 1992), ACT-R (Lewis and Vasishth, 2005), and 4CAPS (Just and Varma, 2007). This work considers activation, and manipulates memory resources by varying the number of analyses the parser considers at each parse step.

Table 1 lists memory factors that may contribute to SUV gradience. They are sensitive to the memory resources available during syntactic parsing, but account for memory differently. Below I describe these variations.

2.2.1 Distance and the DLT

Distance, as measured by the number of words between, for example, a \textit{wh}-word and its verb, has been argued to affect sentence comprehension (Wanner and Maratsos, 1978; Rambow and Joshi, 1994; Gibson, 1998). Experimental evidence supports this claim, but there exist a number of anomalous results that resist explanation in terms of distance alone (Gibson, 1998; Hawkins, 1999; Gibson, 2000). For example, it is not the case that processing difficulty increases solely as a function of the number of words in a sentence. However, it is possible that SUV gradience could be affected by this simple metric.

The Dependency Locality Theory (DLT) (Gibson, 2000) is a more linguistically-informed measure of distance. The DLT argues that an accurate model of sentence processing difficulty is sensitive to the number and discourse-status (given or new) of nominal intervenors that occur across a particular distance. The DLT’s sensitivity to discourse-novelness integrates aspects of D-linking: \textit{which} book, for example, requires that books already be a part of the discourse, though \textit{what} does not (Gundel et al., 1993; Warren and Gibson, 2002). The DLT has been demonstrated to model difficulty in ways that simple distance alone can not (Grodner and Gibson, 2005).

This study also considers a stronger version of the DLT, Intervenors. Intervenors considers both the number and part-of-speech (POS) of nominal intervenors between a \textit{wh}-word and its head. This feature is sensitive to nuanced differences between nominal intervenors, providing a more accurate model of the Memory Facilitation Hypothesis.

2.2.2 Stack memory

Distance can also be measured in terms of the parser’s internal resources. The computational model described here incorporates a stack memory. Although stacks are not accurate models of human memory (McElree, 2000), this architectural property may provide insight into how memory affects SUV gradience.

2.2.3 Activation and interference

Sentence processing difficulty has been attributed to the amount of time it takes to retrieve a word from memory. Lewis & Vasishth (2005) find support for this argument by applying equations from a general cognitive model, ACT-R (Adaptive Control of Thought-Rational) (Anderson, 2005), to a sentence processing model. Their calculation of retrieval time, henceforth retrieval, is sensitive to a word’s activation and its similarity-based interference with other words in memory (Gordon et al., 2002; Van Dyke and McElree, 2006). Activation, Interference, and the conjunction of the two in the form of Retrieval, are considered in this work.

The grammatical and memory-based accounts described above offer several explanations for SUV gradience. They can be represented along a continuum, where the type of information consid-\(^2\)Recent work by Hofmeister and colleagues attributes the advantage to a decrease in memory interference rather than retrieval facilitation (Submitted), but the spirit of the work remains the same.
Hypothesis Sensitive to
---
Distance String distance between words.
DLT Number of nominal intervenors.
Intervenors POS of nominal intervenors.
Stack Memory Elements currently in parser memory.
Baseline Activation Amount structure is activated in memory.
Interference Amount of competition from similar words in memory.
Retrieval Retrieval time of word from memory.

Table 1: Memory-based sentence processing theories.

The computational model not only formalizes the memory accounts, but also provides a framework for memory-based factors that require a computational model, such as retrieval. The results determine memory factors that best account for SUV gradience patterns.

3 Methodology

The test set for SUV gradience is the experimental results from Arnon et al. (To appear). The experiment tests gradience across four conditions, shown in Examples (5)-(8).

(5) Pat wondered what who read. (bare.bare)
(6) Pat wondered what which student read. (bare.which)
(7) Pat wondered which book who read. (which.bare)
(8) Pat wondered which book which student read. (which.which)

The conditions substitute the wh-type of both wh-fillers and wh-intervenors in the island context. In Example (5) both the filler and intervener are bare (the bare.bare condition), whereas in Example (8), both the filler and intervener are which-Ns (which.which). Examples (6) and (7) provide the other possible configurations.

Arnon and colleagues find which.which to be the fastest condition. Figure 1 depicts these results. The other conditions are more difficult, at varying levels: the which.bare condition is less difficult than the bare.which condition, and both are less difficult than the bare.bare condition. These results roughly pattern with acceptability judgments discussed in syntactic literature (Pesetsky, 1987).

Corpora for superiority processing results do not exist. Further, few studies on SUVs incorporate the same structures, techniques, and experimental conditions. Although Arnon et al. considered 20 lexical variations, the unlexicalized parser can not distinguish these variations. Therefore, the parser is only evaluated on these four sentences; however, they are taken to represent classes of structures that generalize to all SUV gradience in English.

3.1 The parsing model

The computational model is based on Nivre’s (2004) dependency parsing algorithm. The algorithm builds directed, word-to-word analyses of test input following the Dependency Grammar syntactic formalism (Tesnière, 1959; Hays, 1964). Figure 2 depicts the full dependency analysis of the which.which condition from Example (8).
Figure 2: A dependency analysis of the which.which condition.

(8), where heads point to their dependents via arcs.

The Nivre parser assembles dependency structure incrementally by passing through parser states that aggregate four data structures, shown in Table 2. The stack $\sigma$ holds parsed words that require further analysis, and the list $\tau$ holds words yet to be parsed. $h$ and $d$ encode the current list of dependency relations.

<table>
<thead>
<tr>
<th>Original POS</th>
<th>Wh</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>WP</td>
<td>WP-WHAT</td>
<td>what</td>
</tr>
<tr>
<td>WP</td>
<td>WP-WHO</td>
<td>who</td>
</tr>
<tr>
<td>WDT</td>
<td>WDT-WHICH</td>
<td>which book</td>
</tr>
<tr>
<td>WDT</td>
<td>WDT-WHAT</td>
<td>what book</td>
</tr>
<tr>
<td>IN</td>
<td>IN-WHETHER</td>
<td>whether</td>
</tr>
<tr>
<td>WRB</td>
<td>WRB</td>
<td>how/why/when</td>
</tr>
</tbody>
</table>

Table 3: Parser configuration.

The parser transitions from state to state via four possible actions: Shift and Reduce manipulate $\sigma$. LeftArc and RightArc build dependencies between $\sigma_1$ (the element at the top of the stack) and $\tau_1$ (the next input word); LeftArc makes $\sigma_1$ the dependent, and RightArc makes $\sigma_1$ the head.

The parser determines actions by consulting a probability model derived from the Brown Corpus (Francis and Kucera, 1979). The corpus is converted to dependencies via the Pennconverter tool (Johansson and Nugues, 2007). The parser is then simulated on these dependencies, providing a corpus of parser states and subsequent actions that form the basis of the training data. Because the parser is POS-based, this corpus is manipulated in two ways to sensitize it to the differences in the experimental conditions. First, the corpus is given finer-grained POS tags for each of the wh-words, described in Table 3.

Secondly, which-N dependencies are encoded as DPs (determiner phrases) and are headed by the wh-phrase (Abney, 1987). This ensures the parser differentiates a wh-word retrieval from a simple noun retrieval, which is necessary for several of the memory-based constraints. Other noun phrases are headed by their nouns. The corpus is not switched to a fully DP analysis to preserve as many of the original relationships as possible.

I extend the Nivre algorithm to allow for beam search within the parser state space. This allows the parser to consider different degrees of parallelism $k$, and manipulate the amount of memory allotted to incremental parse states. This manipulation serves as a model of variation in an individual’s memory as a sentence is parsed.

3.2 Evaluation

To determine how well the accounts model the experimental data, I consider the likelihood of the parser resolving the island-violating dependency between wh-fillers and their verbs in the Arnon et al. data. In terms of the dependency parser, the test determines whether the parser creates a LeftArc attachment in a state where which or what is $\sigma_1$ and read is $\tau_1$. The dependency structure associated with this parser state is depicted in Figure 3 for the which.which condition.

This evaluation is categorical rather than statistical: SUV-processing is based on the decision to form an attachment in a superiority-violating context, given four experimental sentences. While future work will incorporate more experiments for robust statistical analysis, this work focuses on a small subset that generalizes to the greater phenomenon.

3.3 Encoding constraints

The parser determines actions on the basis of probabilistic models, or features. In this work, I en-
code each of the grammatical and memory-based explanations as its own feature. I normalize the weights from the LIBLINEAR (Lin et al., 2008) SVM classification tool to determine probabilities for each parser action (\textit{LeftArc, RightArc, Shift, Reduce}). The features are sensitive to specific aspects of the current parser state, allowing an examination of whether the features suggest the superiority violating \textit{LeftArc} action in the context depicted in Figure 3. The prediction is that attachment will be easiest in the which.which condition and impossible in the other conditions when memory resources are limited \((k=1)\), as in Table 4.

Table 4: \textit{LeftArc} attachments given Arnon et al. (To appear) results. \(Y = \text{Yes}, \ N = \text{No}\).

<table>
<thead>
<tr>
<th>Condition</th>
<th>b.b</th>
<th>b.w</th>
<th>w.b</th>
<th>w.w</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attachment</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
</tbody>
</table>

Table 6: POS for nominal intervenors.

The sequence of \textit{STACKNEXT} features are sensitive to the parser’s memory, in the form of the POS of elements at varying depths of the stack. These features are found to have high overall accuracy in the Nivre parser (Nivre, 2004) and in human sentence processing modeling (Boston et al., 2008).

**Activation, Interference, and Retrieval** predictions are based on the sequence of Lewis & Vasishth (2005) calculations provided in Equations 1-4. These equations require some notion of duration, which is calculated as a function of parser actions and word retrieval times. Table 7 describes this calculation, motivated by the production rule time in Lewis & Vasishth’s ACT-R model.

<table>
<thead>
<tr>
<th>Transition</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEFT</td>
<td>50 ms + 50 ms + Retrieval Time</td>
</tr>
<tr>
<td>RIGHT</td>
<td>50 ms + 50 ms + Retrieval Time</td>
</tr>
<tr>
<td>SHIFT</td>
<td>50ms</td>
</tr>
<tr>
<td>REDUCE</td>
<td>0ms</td>
</tr>
</tbody>
</table>

Table 7: How time is determined in the parser.

Because only words at the top of the stack can be retrieved, the following will be described for \(\sigma_1\). Retrieval time for \(\sigma_1\) is based on its activation \(A\), calculated as in Equation 1.

\[
A_i = B_i + \sum_j W_j S_{ji} \tag{1}
\]

Total activation is the sum of two quantities, the word's baseline activation \(B_i\) and similarity-based interference for that word, calculated in the second addend of the equation. The baseline activation, provided in Equation 2, increases with more
recent retrievals at time $t_j$. This implementation follows standard ACT-R practice in setting the decay rate $d$ to 0.5 (Lewis and Vasishth, 2005; Anderson, 2005).

$$B_i = \ln \left( \sum_{j=1}^{n} t_j^{-d} \right)$$  \hspace{1cm} (2)

$\sigma_1$’s activation can decrease if competitors, or other words with similar grammatical categories, have already been parsed. In Equation (1), $W_j$ denotes weights associated with the retrieval cues $j$ that are shared with these competitors, and $S_{ji}$ symbolizes the strengths of association between cues $j$ and the retrieved item $i$ ($\sigma_1$). For this model, weights are set to 1 because there is only one retrieval cue $j$ in operation: the POS. The strength of association $S_{ji}$ is computed as in Equation 3.

$$S_{ji} = S_{\text{max}} - \ln(\text{fan}_j)$$  \hspace{1cm} (3)

The fan, $\text{fan}_j$, is the number of words that have the same grammatical category as cue $j$, the POS. The maximum degree of association between similar items in memory is $S_{\text{max}}$ which is set to 1.5 following Lewis & Vasishth.

To get the retrieval time, in milliseconds, of $\sigma_1$, the activation value calculated in Equation 1 is inserted in Equation 4. The implementation follows Lewis & Vasishth in setting $F$ to 0.14.

$$T_i = Fe^{-A_i}$$  \hspace{1cm} (4)

The time $T_i$ is the quantity the parser is sensitive to in determining attachments based on the RETRIEVAL feature. Because it is possible that SUVs are better modeled by only part of the retrieval equation, such as baseline activation or interference, the implementation also considers ACTIVATION and INTERFERENCE features. The features are sensitive to the quantities in the addends in Equation 1, $B_i$ and $\sum_j W_j S_{ji}$ respectively.

### 4 Results

The results focus on whether the parser chooses a LeftArc attachment when it is in the configuration depicted in Figure 3 given the grammatical and memory constraints listed in Table 5. Table 8 depicts the outcome, where $Y$ signifies a LeftArc attachment is preferred and $N$ that it is not.

Only one feature correctly patterns with the experimental evidence: INTERVENORS. It allows a LeftArc in the which.which condition, and disallows the arc in other conditions. The INTERVENORS feature also patterns with the experimental evidence as more memory is added. Table 9 depicts the LeftArc attachment for increasing levels of $k$ with this feature. At $k=1$, the parser only chooses the attachment for the which.which condition. At $k=2$, the parser chooses the attachment for both which.which and which.bare. At $k=3$, it chooses the attachment for all conditions. This mimics the decreases in difficulty evident in Figure 1, and provides support for reductionist theories: if memory is restricted ($k=1$), only the easiest attachment is allowed. As memory increases, more attachments are possible.

INTERVENORS is sensitive to the nominal in-

<table>
<thead>
<tr>
<th>Feature</th>
<th>Feature Type</th>
<th>Includes</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELMIN</td>
<td>Yes/No</td>
<td>$\sigma_1 \text{wh-word} \Leftrightarrow \text{intervenors}_\text{wh-word} (\sigma_1 \cdots \tau_1)$</td>
</tr>
</tbody>
</table>

### Table 5: Feature specification. :: indicates concatenation.
### Table 8: LeftArc attachments for the experimental data.

<table>
<thead>
<tr>
<th>Condition</th>
<th>b.b</th>
<th>b.w</th>
<th>w.b</th>
<th>w.w</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Grammar</th>
</tr>
</thead>
<tbody>
<tr>
<td>REL.MIN=YES</td>
</tr>
<tr>
<td>REL.MIN=NO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISTANCE</td>
</tr>
<tr>
<td>DLT</td>
</tr>
<tr>
<td>INTERVENORS</td>
</tr>
<tr>
<td>STACK1NEXT</td>
</tr>
<tr>
<td>STACK2NEXT</td>
</tr>
<tr>
<td>STACK3NEXT</td>
</tr>
<tr>
<td>ACTIVATION</td>
</tr>
<tr>
<td>INTERFERENCE</td>
</tr>
<tr>
<td>RETRIEVAL</td>
</tr>
</tbody>
</table>

### Table 9: INTERVENORS allows more attachments as k increases.

<table>
<thead>
<tr>
<th>Condition</th>
<th>h.b</th>
<th>h.w</th>
<th>w.b</th>
<th>w.w</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERVENORS k=1</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>INTERVENORS k=2</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>INTERVENORS k=3</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

The positive results from the INTERVENORS feature confirms the discourse accessibility hierarchy encoded in the DLT (Gundel et al., 1993; Warren and Gibson, 2002), but only when *wh*-words are included as nominal intervenors. The results also suggest that it is the type, and not just the number of intervenors as suggested by the DLT, that is important.

Further, the INTERVENORS feature does not pattern with the DLT hypothesis. DLT assumes that increasing the number of nominal intervenors causes sentence processing difficulty (Gibson, 2000; Warren and Gibson, 2002). Here, the number of intervenors is increased, but sentence processing is relatively easier. This effect is explained by the intrinsic difference between the DLT and INTERVENORS features: INTERVENORS provides more information to the parser, in the form of the POS of all intervenors. This indicates that certain intervenors help, rather than hinder, the retrieval process.

The negative results demonstrate that other representations of memory do not model SUV gradience. If we consider this along the continuum from (4), those features that take into account less information than INTERVENORS (DISTANCE and DLT) are too restrictive. Of those features that are more complex than INTERVENORS, many are too permissive, or permit the wrong attachments. This pattern is also visible in the STACKNEXT features: STACK1NEXT is too restrictive, while STACK3NEXT too permissive. STACK2NEXT unfortunately permits the wrong attachments. This pattern in the continuum indicates that an intermediate amount of memory information is required to adequately model these results.

INTERFERENCE, which also considers competitors in the intervening string, would seem likely to pattern with the INTERVENORS results. In fact, similarity-based interference and retrieval have previously been argued to account for these gradience patterns (Hofmeister et al., Submitted). However, the only words considered as competitors with *which* for both features in this model are other *wh*-words. For the which.which condition, for example, INTERFERENCE would only consider the second *which* a competitor. INTERVENORS, on the other hand, considers *book*,
which, and student as possible intervenors. This suggests that the INTERFERENCE measure in retrieval would be more accurate if it considered more competitors, a consideration for future work.

Hofmeister (2007) suggests that it is not a single memory factor, but a number of factors, that contribute to SUV gradience. Some features, such as INTERFERENCE or DLT, may be more accurate when they are considered in addition to other features. It is also likely that probabilistic models that include many features will be more robust than single-feature models, particularly when tested on similar phenomena, like islands. I leave these possibilities to future work.

Although the variable beam-width INTERVENORS feature patterns well with the Arnon et al. results, it does not capture the reading time difference between the bare.bare and the bare.which conditions; both are unavailable at \( k=2 \) and available at \( k=3 \). Although this may indicate a problem with the feature itself, it is also possible that a more gradient evaluation technique is needed. As suggested in Section 4, determining accuracy on the basis of attachment alone may be insufficient to correctly model the full experimental evidence in terms of reading times. This is an empirical question that can be tested with this computational model. In future work, I consider the role of parser difficulty, via linking hypotheses such as surprisal, in modeling the experimental data.

The interpretation of Relativized Minimality used here as a grammatical constraint could not derive the experimental results. LeftArc is not preferred when the parser is in a SUV context (RELMIN=Yes)—an expected result as attachments should not occur in SUV contexts. However, the which which, which bare, and the bare which conditions are not violations because they include non-referential NPs. Even with the RELMIN=NO feature, the parser does not select LeftArc attachments, suggesting grammatical gradience is not useful in modeling the SUV gradience results.

This model does not attempt to capture experimental evidence that SUVs and similar phenomena, like islands, are better modeled by grammatical constraints (Phillips, 2006; Sprouse et al., Submitted). Not only does this work only focus on one kind of grammatical constraint for SUV gradience, but the results reported here do not reveal whether the intervention effect itself is better modeled by grammatical or reductionist factors. Rather, the results demonstrate that the gradience in the intervention effect is better modeled by memory than by the gradient grammatical feature. Future work with this computational model will allow for an examination of those memory factors and grammatical factors most useful in exploring the source of the intervention effect itself.

6 Conclusion

This study considers grammatical and memory-based explanations for SUV gradience in a human sentence processing model. The results suggest that gradience is best modeled by a parser that can vary memory resources while being sensitive to the types of nominal intervenors that have been parsed. Grammatical and other memory constraints do not determine correct attachments in the SUV environment. The results argue for a theory of language that accounts for SUV gradience in terms of specific memory factors.

Acknowledgments

I am grateful to Sam Epstein, John T. Hale, Philip Hofmeister, Rick Lewis, Colin Phillips, students of the University of Michigan Rational Behavior and Minimalist Inquiry course, and two anonymous reviewers for helpful comments and suggestions on this work.

References


