

AN EMPIRICAL EVALUATION OF SENTENCE PROCESSING MODELS: CENTER EMBEDDINGS IN HINDI

Shravan Vasishth

Abstract

Data from Hindi center-embedding constructions (CECs) are used to evaluate three sentence processing models: Joshi's Embedded Pushdown Automaton (EPDA), Gibson's Syntactic Prediction Locality Theory (SPLT), and Lewis' Interference and Confusability Theory (ICT). The SPLT and ICT (but not the EPDA) are found to correctly predict several processing facts about Hindi. However, the experimental results also reveal a problem for these two current, wide-coverage theories: neither model appears to be able to account for differences in reading time observed at noun phrases in Hindi CECs. A sentence processing model is proposed in an accompanying article (see (Vasishth & Kruijff 2001) in this volume) that can in principle be integrated with the ICT to provide a unified account of processing difficulty in the languages investigated.

1 Introduction

Several cross-linguistically applicable models of sentence processing have been proposed over the last decade that attempt to account for processing difficulties experienced by humans. Center-embedding constructions (described below in detail) have been

a centerpiece, so to speak, of these models. In this chapter, I discuss the predictions of three models using center embeddings in Hindi, and show that these models make several incorrect predictions regarding the Hindi data. In response to this gap between the data and the existing theories, Vasishth and Kruijff present a model of processing (see the article accompanying this one (Vasishth & Kruijff 2001)); this model can account for the Hindi data, as well as the existing set of data available for Dutch, German, and Japanese center embeddings.

I begin by describing the performance issues relating to center embeddings in general. Then I present three models of sentence processing (developed by Joshi, Gibson and Lewis) and their respective predictions for Hindi. Finally, I evaluate these models using new experimental data from Hindi.

2 What are center embeddings and why are they interesting?

Center-embedding constructions (CECs) involve sentences in which linguistic material is embedded inside another clause. An example is the center embedding (1a), which has one embedded clause. Chomsky & Miller (1963), among others, have observed that double center embeddings like (1b), which have two embedded clauses, are more difficult for English native speakers to process than single embeddings (1a) or right-embedded constructions like (1c).

- (1) a. The rat [that the cat chased] ate the malt.
 b. The rat [that the cat [that the dog chased] killed] ate the malt.
 c. The dog chased the cat [that killed the rat [that ate the malt]].

A widely-held view is that limitations on human working memory¹ impose strong constraints on the processing of complex structures like CECs. The assumption is that the noun phrases must be temporarily stored in working memory until verbal information clarifies the sentence structure. Two wide-coverage theories of sentence processing, Gibson's Syntactic Prediction Locality Theory (SPLT) (Gibson 1998; Babyonyshev & Gibson 1999), and Lewis' Interference and Confusability Theory (ICT) (Lewis 1998), specifically appeal to working memory constraints in explaining the processing of syntactic structures like CECs. Joshi's Embedded Pushdown Automaton (EPDA) does not appeal to working memory constraints directly, but it does rely on the notion of temporary storage of material. Gibson and Lewis' models are able to account for many processing facts in languages such as Dutch (Kaan & Vasić 2000), German (Bach *et al.* 1986), Japanese (Nakatani *et al.* 2000), and Korean (Uehara & Bradley 1996), and Joshi's can do the same for a smaller range of languages.

¹I assume that working memory, or short-term memory, is "... a short-duration system in which small amounts of information are simultaneously stored and manipulated in the service of accomplishing a task" (Caplan & Waters 1999).

Clearly, many other languages need to be investigated before theories of sentence processing can claim truly universal coverage (as these models aspire to do). This is the motivation for studying the processing of CECs in Hindi.² This is a useful language to investigate since it has certain properties not seen in previously studied languages. We look at these properties next. Consider first the single center embedding in example (2):

- (2) Siitaa-ne Hari-ko [kitaab(-ko) khariid-ne-ko] kahaa
 Sita-erg Hari-dat book(-acc) buy-inf told
 ‘Sita told Hari to buy the book.’

Here, the ergative case marker *-ne* marks the agent,³ and the other noun phrases (NPs) are marked by the oblique case marker *-ko*, regardless of the NP’s grammatical role as indirect or direct object. However, case marking on the direct object (*kitaab*) is optional: when present, it marks the NP as specific, and when absent, the NP could be specific or non-specific (Mohanani 1994).

For example, in a sequence of utterances like (3), the direct object *kitaab* cannot have case marking when it is not salient in the discourse (3a), but can have it once it has been mentioned (3b).

- (3) a. Siitaa aura Hari-ne dukaan mē ek kitaab dekhii
 Sita and Hari-erg shop in one book saw
 ‘Sita and Hari saw a book in a shop.’
 b. Sita-ne Hari-ko [kitaab(-ko) khariid-ne-ko] kahaa
 Sita-erg Hari-dat book-acc buy-inf told
 ‘Sita told Hari to buy the book.’

The interesting fact for us is that, in example (2), if *-ko* case marking is present on the direct object (*kitaab*), the second and third NPs will have phonologically the same suffix. This is interesting because previous research on adjacent similarly case-marked NPs in Japanese and Korean CECs Uehara & Bradley (1996); Lewis & Nakayama (1999) have shown that nominative case marking on adjacent NPs results in increased processing difficulty, presumably due to working memory overload (this is discussed in detail below). However, it is an open question whether case markings other than nominative affect processing similarly.

Hindi also has rather free word order in general; there is only one constraint on the 5! orders for the single center embedding in (2): the direct object of the most deeply

²Hindi, also known as Urdu, or Hindi-Urdu, is an Indo-Aryan language spoken primarily in South Asia; it has about 424 million speakers in India (source: 1991 Census of India, www.censusindia.net), and about 10 million in Pakistan (source: www.sil.org).

³Hindi is a split-ergative language, with an ergative-absolutive case marking system in the perfective aspect.

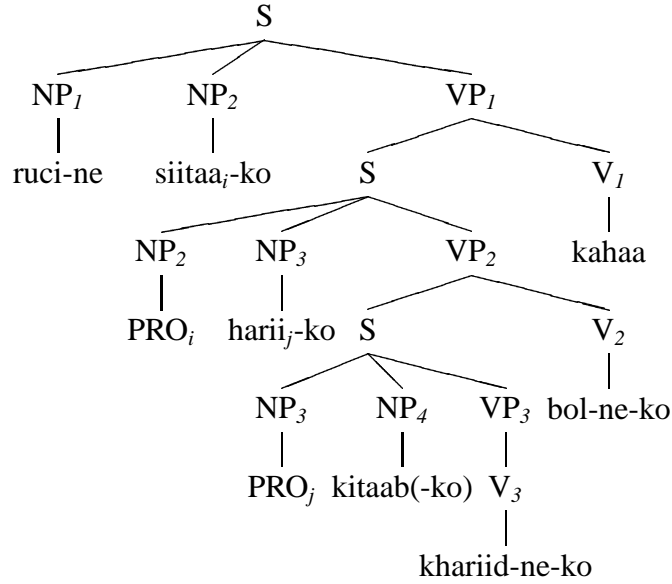


Figure 1: Example (5)

embedded verb must not appear to the right of this verb, as the following example shows (unlike analogous English sentences like *The cat the dog bit died*, examples like (4a) are very natural in Hindi and occur quite frequently in a large text corpus (Vasishth *et al.*)).

- (4) a. Siitaa-ne Hari-ko [kitaab khariid-ne-ko] kahaa
 Sita-erg Hari-dat book buy-inf told
 ‘Sita told Hari to buy a/the book.’
 b. * Siitaa-ne Hari-ko [khariid-ne-ko kitaab] kahaa
 Sita-erg Hari-dat buy-inf book told
 ‘Sita told Hari to buy a/the book.’

This near-absence of constraints on word order turns out to be very useful in evaluating the existing models of sentence processing, as we shall presently see.

A third property of Hindi center embeddings is that these are control constructions. That is, the structure of a double embedding like (5) is as shown in Figure 1 (single embeddings have a similar structure).

- (5) Ruci-ne Siitaa-ko [Hari-ko [kitaab(-ko) khariid-ne-ko] bolne-ko] kahaa
 Ruci-erg Sita-dat Hari-dat book(-acc) buy-inf tell-inf told
 ‘Ruci told Sita to tell Hari to buy the book.’

That is, the indirect object of a clause at a given level (matrix or embedded) obligatorily controls a PRO in subject position in the clause embedded within it. The syntax of these

constructions is discussed in detail elsewhere (Vasishth in progress).

These three properties (phonologically similar case marking with dative and accusative case, relatively free word order, and center embeddings being control constructions) become relevant as we look at Hindi CECs to test the predictions of the EPDA, SPLT, and ICT. I will show that the SPLT and ICT can only partly account for the Hindi processing facts and that the EPDA fails almost completely. Specifically, Gibson's SPLT can only partly account for certain reading time differences for NPs. On the other hand, Lewis' ICT appears to be noncommittal about NP reading time differences: it assumes that the primary source of processing difficulty for CECs occurs in the retrieval stage,⁴ as NPs stored in working memory are retrieved and integrated with information about the verb. However, findings from self-paced reading experiments presented in this paper (see Section 4) indicate an additional, earlier, more prominent source of processing difficulty in the NP encoding/storage⁵ stage. On this basis, I argue that working-memory related constraints on parsing are affected by both encoding and retrieval.

Let us now turn to the three sentence processing models in question.

3 Three models of sentence processing

3.1 Joshi's Embedded Pushdown Automaton (1990)

Joshi (Joshi 1990) presents a computational model of processing based on the results of (Bach *et al.* 1986); the latter paper showed that Dutch crossed dependencies were easier to process for native Dutch speakers than German nested dependencies are for native German speakers. Examples of crossed Dutch and nested German dependencies are shown below:

- (6) a. Jan Piet Marie zag laten zwemmen
 Jan Piet Marie saw make swim
 'Jan saw Piet make Marie swim.'
 NP1 NP2 NP3 V1 V2 V3
- b. ...dass Hans Peter Marie schwimmen lassen sah
 ...that Hans Peter Marie swim make saw
 '... that Hans saw Peter make Marie swim.'
 NP1 NP2 NP3 V3 V2 V1

⁴By 'retrieval' I mean the *process* of integration of NP information with a verb.

⁵I use the term 'encoding' to refer to the stage preceding storage of NPs in working memory whereby the NPs are converted into some representational form suitable for storage. Gathercole & Baddeley (1993) present a discussion relating to the working memory processes assumed here.

The Dutch CECs are called “crossed” because of the fact that the dependencies between the verbs and the subjects form crossing chains (NP1 NP2 NP3 V1 V2 V3), and the German CECs are nested since the pattern is NP1 NP2 NP3 V3 V2 V1.

(Bach *et al.* 1986) “...show that the pushdown automaton (PDA) cannot be the universal basis for the human parsing mechanism” (Joshi 1990). The problem for the PDA is that in the case of German, NP3 and the immediately following V3 can combine together, but there is no way to tell where that structure belongs until one gets to the end of the sentence, and so this structure (and, similarly, the NP2-V2-(NP3-V3) sub-structure) has to be stored until a higher structure becomes available. By contrast, in Dutch, the sub-structures can be built and integrated incrementally.

Joshi proposes a PRINCIPLE OF PARTIAL INTERPRETATION to overcome this problem with PDAs. As he puts it (Joshi 1990:4-5):

1. The structure should be a properly integrated structure with respect to the predicate-argument structure (i.e., only predicates and arguments that go together should be integrated: *ad hoc* packaging of predicates and arguments is disallowed), and there should be a place for it to go, if it is expected to fit into another structure (i.e., the structure into which it will fit must have been popped already).
2. If a structure which has a slot for receiving another structure has been popped, then the structure that will fill this slot will be popped next.

Joshi then develops an embedded PDA (EPDA) and shows that it can handle the Dutch and German processing facts. The significance of this is that EPDAs are equivalent to the syntactic formalisms TAGs, HPSG, and CCG, all of which are capable of providing syntactic analyses for crossed and nested dependencies.

In the following discussion of Joshi’s model, I assume that the reader has a working knowledge of PDAs (see, e.g., (Hopcroft & Ullman 1979:107-124) for details). In an EPDA, the pushdown store is a sequence of stacks, and new stacks may be created above or below (to the left or right) of the current stack. The specific behavior of EPDAs described below is based on (Joshi 1990).

1. **Stack head:** This is always at the top symbol of the top stack. If the stack head ever reaches the bottom of a stack, then the stack head automatically moves to the top of the stack below (or to the left) of the current stack, if there is one.
2. **Transition function δ' :** Given an input symbol, the state of the finite control and the stack symbol, this specifies (a) the new state; (b) whether the current stack is pushed or popped; and (c) new stacks to be created above or below the current stack.

δ' (input symbol, current state, stack symbol) =
(new state, sb_1, sb_2, \dots, sb_m , push/pop on current stack, st_1, st_2, \dots, st_n)

where sb_1, sb_2, \dots, sb_m are the stacks introduced below the current stack, and st_1, st_2, \dots, st_n are the stacks introduced above it.

Note that during each move, push/pop is carried out on the current stack, and pushes on the newly created stack(s).

Next, I illustrate processing of the Dutch crossed dependency sequence: NP1 NP2 NP3 V1 V2 V3 with the figure below showing the various states. The column “Stack sequence” contains the newly created stacks, “Stack” is the stack we begin with, and the column “Pop action” shows how the interpretation is incrementally built up. Finally, “No. of (input) items” lists a number that Joshi uses as a complexity measure to account for the difference in processing Dutch and German– this just involves adding up the total number of input items in the EPDA at each move, and looking at the largest number (in the Dutch case, 3).

Input head at	Stack sequence	Stack	Pop action	No. of items
NP1				0
NP2		NP1		1
NP3		NP1 NP2		2
V1		NP1 NP2 NP3		3
V1		NP1 NP2		3
V1	NP3	NP1		3
V1	NP3 NP2			3
V2	NP3 NP2		V1(NP1,S1)	2
V3	NP3		V2(NP2,S2)=S1 V3(NP3)=S2	1 0

Figure 2: EPDA processing of Dutch dependencies

The way this proceeds is as follows. First, NP1 is read in and pushed on to the current stack, the same goes for NP2 and NP3. Then NP3, NP2, and NP1 are successively popped out of the current stack and pushed into sequences of stacks at the left of the current stack. Then, each NP is popped out of the EPDA and incrementally builds up the predicate-argument structures starting with V1 up to V3. The complexity never goes beyond 3.

The problematic German case (problematic for PDAs), where the order of NP and V sequences is NP1 NP2 NP3 V3 V2 V1 is handled as shown below. In each case, V^*n is a possibly underspecified structure encoding Vn and its argument(s) (NPn and possibly also S). That is, $V^*3 = V1(NP3)$, $V^*2 = V2(NP2,S2)$, and $V^*1 = V3(N1,S1)$. Note that the maximum number of input items in this case is 6, higher than that in Dutch crossed dependencies.

Input head at	Stack sequence	Stack	Pop action	No. of items
NP1				0
NP2		NP1		1
NP3		NP1 NP2		2
V3		NP1 NP2 NP3		3
V3		NP1 NP2		4
V3	V*3	NP1		5
V3	V*3 V*2			6
V2	V*3 V*2 V*1		V1(NP1,S1)	4
V1	V*3 V*2 V*1		V2(NP2,S2)=S1	1
			V3(NP3)=S2	0

Figure 3: EPDA processing of German dependencies

Joshi also discusses the case of mixed dependencies in German, where the sequences are like NP1 NP2 NP3 V1 V3 V2. The complexity measure for this kind of dependency is claimed to be intermediate between that for crossed and nested dependencies (presumably due to the larger number of total steps involved in mixed dependencies). In such a case, the EPDA behaves exactly like that for nested dependencies in German until we reach V1. Then it must behave like the EPDA for crossed dependencies. A schematic view is shown below:

Input head at	Stack sequence	Stack	Pop action	No. of items
NP1				0
NP2		NP1		1
NP3		NP1 NP2		2
V1		NP1 NP2 NP3		3
V1		NP1 NP2		3
V1	NP3	NP1		3
V1	NP3 NP2			3
V1	NP3 NP2 NP1			3
V3	NP3 NP2		V1(NP1,S1)	2
V2	NP3 V3 NP2			3
	NP3 V3		V2(NP2)=S1	2
	NP3		V3(NP)	1
			NP3	0

Figure 4: EPDA processing of German mixed dependencies

One point to note here is that when V3 is popped out, its argument (NP) is uninstantiated. This only gets instantiated when NP3 is popped out in the final move. Another important point: when the input head is at V2, the preceding V3 has been inserted to the left of NP2 by creating a new stack behind the stack holding NP2, and inserting V2 into

this new stack. These moves are allowed by the EPDA and accord with the PPI.

3.2 Predictions of the EPDA model for Hindi CECs

Joshi’s account raises some interesting questions for Hindi center-embedding constructions. Recall the issue of specificity marking on the direct object, i.e., minimal pairs like the following:

- (7) a. Siitaa-ne Hari-ko [kitaab khariid-ne-ko] kahaa
 Sita-erg Hari-dat book buy-inf told
 ‘Sita told Hari to buy a/the book.’
- b. Siitaa-ne Hari-ko [kitaab-ko khariid-ne-ko] kahaa
 Sita-erg Hari-dat book-acc buy-inf told
 ‘Sita told Hari to buy the book.’
- c. Siitaa-ne Hari-ko [Ravi-ko [kitaab khariid-ne-ko] bol-ne-ko] kahaa
 Sita-erg Hari-dat Ravi-dat book buy-inf tell-inf told
 ‘Sita told Hari to tell Ravi to buy a/the book.’
- d. Siitaa-ne Hari-ko [Ravi-ko [kitaab-ko khariid-ne-ko] bol-ne-ko] kahaa
 Sita-erg Hari-dat Ravi-dat book-acc buy-inf tell-inf told
 ‘Sita told Hari to tell Ravi to buy a/the book.’

Consider now the parses for (7a,b):

Input head at	Stack sequence	Stack	Pop action	No. of items
NP1-ne				0
NP2-ko		NP1-ne		1
NP3-∅/-ko		NP1-ne NP2-ko		2
V2		NP1-ne NP2-ko NP3		3
V1	V*2	NP1-ne		4
	V*2		V1(NP1-ne,S1)	3
			V2(NP2-ko,NP3)=S1	0

Figure 5: Hindi examples (7a,b)

Based on Table 5, it is easy to see that the EPDA predicts the following for Hindi center embeddings:

- No difference in processing difficulty at NP3 with respect to specificity-marking.
- No difference in processing difficulty at V2 in both sentences.

- Greatest difficulty at final (matrix) verb.

I will presently show that none of these productions are borne out.

3.3 Concluding remarks regarding the EPDA model

Consider again the Dutch vs. German contrast. Bach et al. showed that Dutch crossed dependencies are easier to process for Dutch native speakers, but German nested dependencies are harder for German native speakers. The EPDA models moment by moment processing difficulty (Joshi, personal communication), so it would predict that the highest processing cost is at the innermost verb in both the Dutch and German cases, since in the EPDA the most time is spent there and the number of items present in the EPDA at this point is the largest. However, experimental work has shown that this is not true, at least not for Dutch (Kaan & Vasić 2000): in Dutch, as in Hindi, the most costly region seems to be at the final NP.

Moreover, in the EPDA, structure building does not begin until the verbs are reached; until that point, the NPs are simply stored in the stack. NPs, however, generate predictions (see, e.g., (Scheepers *et al.* 1999), and references cited therein), they are not just merely stored in a temporary buffer (presumably EPDA is intended to model working memory). Gibson's SPLT, discussed next, addresses this issue of incremental processing and predictions at the NPs.

In sum, there are several empirical problems in the EPDA model: the inability to predict moment by moment reading times correctly for Hindi and Dutch (there are no reading time studies for German CECs, as far as I know), and the assumption of simple storage of the NPs before the verbs are encountered.

3.4 Gibson's Syntactic Prediction Locality Theory (1998/1999)

Gibson's syntactic prediction locality theory (SPLT) (Gibson 1998; Babyonyshev & Gibson 1999) has a somewhat different processing cost metric than the EPDA. The SPLT has two cost components: INTEGRATION COST and MEMORY COST. Integration cost is the distance between the head-to-be-integrated (e.g., an NP) and the head to which it connects in the current structure (e.g., a verb). This is quantified in terms of the number of discourse referents separating the two heads. Memory cost is the number of all required syntactic heads at a given point. The memory cost for each predicted syntactic head h increases as linguistic material not matching h is processed. The prediction of the top-level predicate (matrix verb) is assumed to be cost-free (since most utterances are headed by a predicate), and for all required syntactic heads other than the top-level predicate, memory cost $M(n) = n$, where n is the number of new discourse referents processed since that syntactic head was

initially predicted.⁶

I illustrate the model’s predictions by giving a derivation for Hindi double embeddings.⁷ In (8), case marking or the absence thereof on NP3 is indicated by \emptyset (no case marking) and *-ko*. In this discussion, I focus on the memory cost alone for ease of exposition; since integration cost is a function of memory cost in the SPLT, the relative processing costs that interest us remain the same.

(8) NP1-ne NP2-ko NP3-ko NP4- \emptyset /ko V3-inf V2-inf V1

Here, the predicted slowest point during real-time processing is over NP4, and no difference between the two variants (NP4- \emptyset versus NP3-*ko*) is predicted.

Prediction	NP1-ne	NP2-ko	NP3-ko	NP4- \emptyset /ko	V3	inf2	V2	inf1	V1
V1	0	0	0	0	0	0	0	0	0
V2	-	M(0)	M(1)	M(2)	M(2)	M(2)	*	-	-
Inf1	-	M(0)	M(1)	M(2)	M(2)	M(2)	M(2)	*	-
V3	-	-	M(0)	M(1)	*	-	-	-	-
Inf2	-	-	M(0)	M(1)	M(1)	*	-	-	-
	---	---	---	---	---	---	---	---	---
	0	0	2	6	5	4	2	0	0

Figure 6: Processing of (8)

However, I will presently show that the slowest reading time is over the final NP only if it has *-ko* marking. Thus, the predictions of the SPLT appear to be only partly correct.

3.5 Lewis’ Interference and Confusability Theory (1998/1999)

This model treats parsing as a short-term memory task. In the context of center-embedding constructions, the central idea is that retrieval at a verb of an NP during real-time processing is affected by two factors: (i) POSITIONAL CONFUSABILITY; and (ii) RETROACTIVE INTERFERENCE (RI) and PROACTIVE INTERFERENCE (PI).

Positional confusability is the probability of correctly retrieving an NP from among a list of NPs seen up to a given point. For example, if NP1 NP2 NP3 NP4 is the list of NPs seen so far, and if NP3 is to be retrieved, the probability of correct retrieval will decrease if NP3 and NP4 are similarly case marked. This decrease in probability is due to

⁶Only finite verbs introduce discourse referents in this model (Gibson, personal communication).

⁷I follow Babyonyshev and Gibson’s derivation for Japanese center-embeddings and assume that the oblique postpositions/case markers for the embedded verbs are also predicted during real time processing; however, nothing hinges on this assumption.

the assumption that item-recall is with respect to the end-points (the first and last item) of a list (independent motivation for this assumption comes from the psychology literature, e.g., (Henson 1999)). If an end-point NP is similar to the item being recalled (in our case, ‘similar’ means similarly case marked), then the probability of correct retrieval decreases. Conversely, if the end-point NP is dissimilarly case marked compared to the NP to be retrieved, the probability of correct retrieval increases (i.e, positional confusability is reduced).

Pro- and retroactive interference are defined as follows. Proactive interference (PI) occurs when the retrieval of an NP that suffers from interference by an NP or NPs preceding the NP to be retrieved. Retroactive interference (RI) is the opposite: the retrieval of an NP suffers from interference from items that follow the NP. There is a great deal of evidence in the psychology literature for PI and RI in intertrial list recall (see, e.g., (Müller & Pilzecker 1900) and (Keppel & Underwood 1962) for some of the earliest findings). It is an assumption of the model that PI and RI occur *within* a list of NPs (see (Humphreys & Tehan 1998), which provides independent evidence for proactive interference within trials).

I now illustrate the model’s behavior.

$$(9) \quad \phi_1\phi_2 \dots \phi_n X \rho_1\rho_2 \dots \rho_m Y$$

If Y is the current word (a verb), and a syntactic relation needs to be established between a constituent projected from Y and a constituent headed by a prior word X (a noun), the total amount of interference at Y depends on the number of similar items intervening between X and Y (RI) and the number of similar items preceding X (PI). ‘Similarity’ is understood to be syntactic similarity, which is determined by the structural role to be assigned to X . For example, if X is to be assigned the structural position of subject, then RI occurs due to all $\rho_1\rho_2 \dots \rho_m$ which could also fill subject positions, and PI occurs due to all $\phi_1\phi_2 \dots \phi_n$ which could also fill subject positions. In addition, positional confusability increases if X and ρ_m or ϕ_1 (i.e., one of the end points) is similar to X . The total amount of retrieval difficulty at Y is the sum of the two kinds of interference and positional confusability. For ease of exposition, I assign simple numerical values to each component of processing cost: e.g., if there are two elements causing RI, then RI=2, if one end-point is increasing positional confusability (POS), then POS=1, etc. In the actual computational implementation, the costs are not necessarily simple integer values.

The predictions for Hindi CECs illustrate the model’s operation. The pattern in (10a) is predicted to be easier than (10b).

- (10) a. NP1-ne NP2-ko NP3-ko NP4- \emptyset V3 V2 V1
 b. NP1-ne NP2-ko NP3-ko NP4-ko V3 V2 V1

The following tables illustrate how the model works. In each table, the first column lists the item to be retrieved (X in the template above) at a particular verb Y , with the items

$\rho_1\rho_2\dots\rho_m$ intervening between X and Y , and the items $\phi_1\phi_2\dots\phi_n$ preceding X . For each Y the Figure lists the cost of RI and PI, and the uparrow (\Uparrow) indicates the item(s) involved in causing RI or PI at retrieval.

Retrieved item	NP1-ne	NP2-ko	NP3-ko	NP4- \emptyset	V3	V2	V1
NP3-ko	ϕ_1	ϕ_2	X	ρ_1	Y POS=0 RI=0 PI=2		
NP2-ko	ϕ_1	X	ρ_1	ρ_2		Y POS=0 RI=1 PI=1	
NP1-ne	X	ρ_1	ρ_2	ρ_3			Y POS=0 RI=2 PI=0

Figure 7: Processing of (10a)

Retrieved item	NP1-ne	NP2-ko	NP3-ko	NP4-ko	V3	V2	V1
NP3-ko	ϕ_1	ϕ_2	X	ρ_1	Y POS=1 RI=1 PI=2		
NP2-ko	ϕ_1	X	ρ_1	ρ_2		Y POS=1 RI=2 PI=1	
NP1-ne	X	ρ_1	ρ_2	ρ_3			Y POS=1 RI=3 PI=0

Figure 8: Processing of (10b)

Here, the retrieval of a subject at a verb results in the other underlying subjects causing RI or PI.

In the next section I show that Lewis' model correctly predicts increased retrieval difficulty at the innermost verb. However, there is another dimension of processing difficulty in such sentences: encoding difficulty of the NPs increases if similarly case-marked NPs are adjacent to each other. Lewis' model is agnostic about processing difficulties at NPs and is thus unable to account for this fact.

We turn now to the experimental evidence from Hindi.

4 Center embeddings in Hindi: Three experiments

The second author of these notes (Vasishth) conducted three experiment to evaluate various predictions of these three models. These experiments were conducted at Jawaharlal Nehru University, New Delhi, India during September 2000. The research was funded partly through the project, “Establishing Ohio State as a Major Center for Language Processing Research, Ohio State Center for Cognitive Science, Department of Linguistics, and Department of Computer and Information Science” and partly by the Department of Linguistics, The Ohio State University (OSU), and was conducted in accordance with the human subjects research protocol number 80B0433 specified by the Human Subjects Institutional Review Board, OSU.⁸

4.1 Experiment 1

4.1.1 Method and materials

Experiment 1 had a 2×2 factorial design, the two factors being level of embedding (single or double; compare (11a,b) and (11c,d)), and absence or presence of case marking on the final NP (compare (11a,c) and (11b,d)). In the test sentences, all but the final NPs were proper names; the final NP was always an inanimate common noun, such as ‘book’ or ‘letter’. This was a paper questionnaire where subjects were asked to rate each sentence on a scale from 1 (completely unacceptable) to 7 (completely acceptable).

- (11) a. Siitaa-ne Hari-ko [kitaab khariid-ne-ko] kahaa
 Sita-erg Hari-dat book buy-inf told
 ‘Sita told Hari to buy a/the book.’
- b. Siitaa-ne Hari-ko [kitaab-ko khariid-ne-ko] kahaa
 Sita-erg Hari-dat book-acc buy-inf told
 ‘Sita told Hari to buy the book.’
- c. Siitaa-ne Hari-ko [Ravi-ko [kitaab khariid-ne-ko] bol-ne-ko] kahaa
 Sita-erg Hari-dat Ravi-dat book buy-inf tell-inf told
 ‘Sita told Hari to tell Ravi to buy a/the book.’
- d. Siitaa-ne Hari-ko [Ravi-ko [kitaab-ko khariid-ne-ko] bol-ne-ko] kahaa
 Sita-erg Hari-dat Ravi-dat book-acc buy-inf tell-inf told
 ‘Sita told Hari to tell Ravi to buy a/the book.’

⁸All comparisons presented hereafter have $p < .05$, unless otherwise stated.

Four lists were prepared in a counterbalanced, Latin Square design, and 32 fillers were inserted between 16 target sentences in pseudorandomized order. The fillers consisted of eight examples of four syntactic structures: relative clauses, medial gapping constructions, simple declaratives, and sentences with that-clauses (all the stimuli and fillers are available from the author on request). Fifty-three native speakers of Hindi participated in the experiment. Nineteen of these were Hindi-speaking students at the Ohio State University, and were paid 5 US Dollars each for completing the questionnaire; the remaining thirty-four were undergraduate and graduate students at Jawaharlal Nehru University, New Delhi, India, and were paid 80 Indian Rupees each (approximately 1.7 US Dollars).

4.1.2 Predictions

This experiment tested the following predictions:

- Acceptability will decrease with increasing level of embedding. All three models predict this.
- Lewis’ model predicts that direct-object marking will result in reduced acceptability, but Gibson’s and Joshi’s models predict that the direct object marking will have no effect on acceptability.

4.1.3 Results

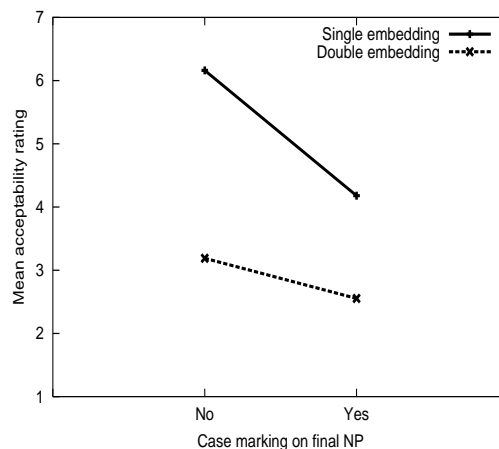


Figure 9: Results of Experiment 1

As Figure 9 shows, the results indicate that increasing the amount of embedding reduces acceptability ((11c,d) were less acceptable than (11a,b)), as predicted by Joshi’s, Gibson’s, and Lewis’ models. However, case marking on the final NP also results in

reduced acceptability ((11b), (11d) were less acceptable than (11a), (11c) respectively), which Lewis' model predicts, but Joshi's and Gibson's do not. The details of the statistical analysis are as follows: A repeated measures analysis of variance (ANOVA) was done for subject (F1) and item (F2) means, with level of embedding and presence or absence of case marking on the final NP as the within-subject factors. The mean rating for sentences like (11a) was significantly higher (mean: 6.162) than that for sentences like (11b) (mean: 4.179), $F(1,52) = 130.969$, rating for sentences like (11c) was significantly higher (mean: 3.189) than for sentences like (11d) (mean: 2.553), $F(1,52) = 13.447$,

4.2 Experiment 2

4.2.1 Method and Materials

This was a noncumulative self-paced moving window reading task (Just *et al.* 1982); exactly the same materials were used as for Experiment 1 (see examples (11) for the four conditions).

A G3 laptop Macintosh running PsyScope (Cohen *et al.* 1993) was used to present the materials to subjects. Forty-six native speakers of Hindi participated in the experiment; no subjects from Experiment 1 participated in this experiment. The task was to press the space key in order to see each successive word; each time the key was pressed, the previous word would disappear. Reading time (msec) was taken as a measure of relative momentary processing difficulty. A yes/no comprehension question was presented after each sentence; these were meant to ensure that subjects were attending to the sentences.

4.2.2 Predictions

This experiment tested the following predictions:

- Reading time at the innermost verb would be slower in examples like (11a,c) than in examples like (11b,d). This was based on Lewis' interference theory, which states that the probability of correct retrieval of the final NP decreases as its positional confusability with an adjacent NP increases.
- Reading time would be slowest either at (i) the last NP (SPLT), or (ii) the innermost verb (EPDA). Prediction (i) is based on the SPLT, as discussed in Section 3.4. Prediction (ii) comes from the fact that in the EPDA processing of examples like (11b,d) will proceed as in German center embeddings (see Figure 3), with a highest cost of 7 at the innermost *verb* (since there is one more NP than in the German example in Figure 3).

- The EPDA and SPLT both predict that reading time over the last NP will be unaffected by whether the NP has case marking or not.

4.2.3 Results

Residual reading time were calculated for each region by subtracting from raw reading times each participant’s predicted reading time for regions with the same numbers of characters; this in turn is calculated from a linear regression equation across all of a participant’s sentences in the experiment (Ferreira & Clifton 1986; Trueswell *et al.* 1994). This was done in order to factor out the effect of word length on reading time. However, the raw reading times gave identical results to the ones discussed below.

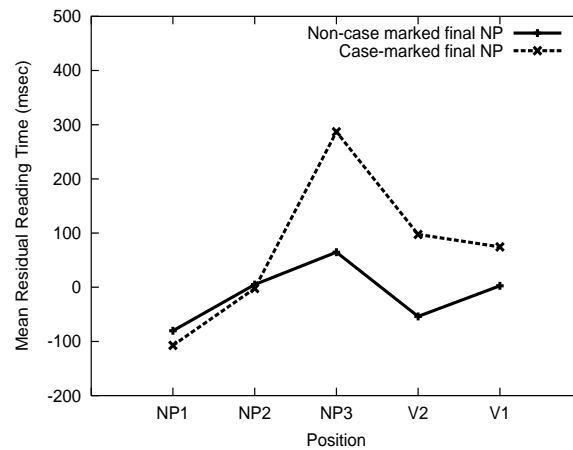


Figure 10: Single Embeddings

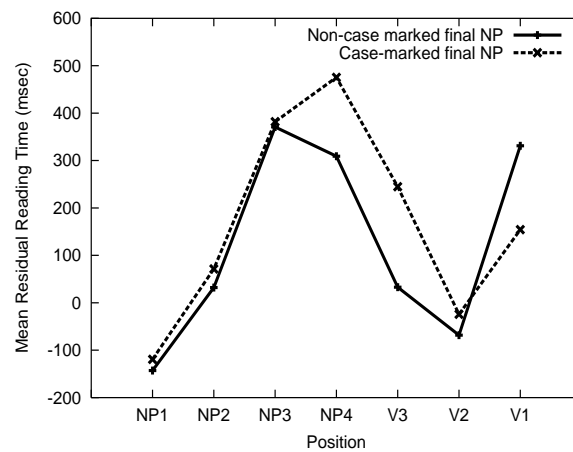


Figure 11: Double Embeddings

As shown in Figures 10 and 11, the results indicate that (a) reading time (RT) increases at the second of two adjacent similarly case-marked NPs; (b) RT remains slow if two *-ko* marked NPs are followed by a third *-ko* marked NP; (c) RT is faster if a non-case-marked NP (rather than a case-marked NP) follows a *-ko* marked NP; (d) RT at the innermost verb is slower if the last NP is case marked than when it isn't; and (e) the slowest RT is in the region of the final NP, particularly if it is case marked.

Thus, the first prediction (Lewis' model), that RTs would be slower at the innermost verb in sentences with case-marked final NPs than in sentences with non-case-marked final NPs, was borne out.⁹ The second prediction (Gibson's model's), that the slowest RT would be at the final NP was partly confirmed, and Joshi's model's prediction that the slowest RT would be at the innermost verb, was disconfirmed. The third prediction (Gibson's and Joshi's models'), that RT at the last NP would be unaffected by case marking, was disconfirmed.

Thus, Lewis' and Gibson's models make several correct predictions. However, both models are unable to capture some of the Hindi facts: Lewis' ICT makes no predictions for the NP reading times,¹⁰ and Gibson's model cannot account for the different RTs on the final case-marked vs. non-case-marked NPs. Thus, it is clear that encoding/storing NPs is a component of processing that neither model can account for.

I propose to extend Lewis' model so that it can account for encoding and retrieval difficulty; this is discussed in (Vasishth & Kruijff 2001). I choose to augment Lewis' model rather than Gibson's because the former makes no assumptions about the encoding component of processing and it is straightforward to incorporate the ideas set forth in (Vasishth & Kruijff 2001), which provides a fairly robust account of difficulty due to encoding processes.

We now consider another aspect of Lewis' model. Recall that Lewis identifies two sources of retrieval difficulty: positional confusability and interference (Section 3.5). In experiment 2, there was no way to distinguish between the two. In experiment 3 below, I attempt to find evidence for positional confusability. I use the fact that positional confusability predicts that processing will improve if similarly case-marked NPs are made non-adjacent, by, e.g., scrambling. I therefore tested this prediction in Experiment 3 by manipulating adjacency.

⁹It is possible that the longer reading time at the innermost verb is due to spillover to the verb region from processing difficulty at the NPs. I intend to investigate this question further in future research.

¹⁰Lewis (personal communication) informs me that this claim is incorrect; Lewis' ICT does indeed make predictions for NP reading times. However, at the time of writing this I do not possess a description of the precise predictions made by the ICT.

4.3 Experiment 3

4.3.1 Method and Materials

This was an offline acceptability rating task similar in design to Experiment 1. The test sentences were single embeddings; one factor was presence or absence of case-marked final NPs, and the other factor was scrambled (NP2-**ko** NP1-*ne* NP3(-**ko**)) or unscrambled (NP1-*ne* NP2-**ko** NP3(-**ko**)) first and second NPs (see examples (12a,b)).

Participants were given a paper questionnaire and asked to rate each sentence on a scale of 1 (completely unacceptable) to 7 (completely acceptable). Sixty-seven native speakers of Hindi participated; none had participated in the earlier experiments. There were 16 test items and 32 fillers.

- (12) a. siitaa-ne hari-ko kitaab khariid-ne-ko kahaa
 Sita-erg Hari-dat book buy-inf told
 ‘Sita told Hari to buy a/the book.’
- b. hari-ko siitaa-ne kitaab khariid-ne-ko kahaa
 Hari-dat Sita-erg book buy-inf told
 ‘Sita told Hari to buy a/the book.’
- c. siitaa-ne hari-ko kitaab-(ko) khariid-ne-ko kahaa
 Sita-erg Hari-dat book-acc buy-inf told
 ‘Sita told Hari to buy the book.’
- d. hari-ko siitaa-ne kitaab-(ko) khariid-ne-ko kahaa
 Hari-dat Sita-erg book-acc buy-inf told
 ‘It was Hari who Sita told to buy the book.’

The conditions (12a) and (12b) were included to establish whether scrambled sentences are in general less acceptable than unscrambled ones when presented out of context. It is well-known that scrambled sentences (presented out of context) are less acceptable in languages like English, German, Finnish, and, Hungarian, (see (Hyönä & Hujanen 1997) for a discussion and references). We would therefore expect scrambled sentences (in null contexts) to be involve some processing cost. One key question is whether positional confusability has a greater cost compared to the processing cost of scrambling. If increasing positional confusability has a higher relative cost than scrambling, we will have evidence consistent with the confusability theory.

4.3.2 Predictions

Scrambling was expected to result in reduced acceptability; in addition, adding case marking to the final NP in a scrambled sentences is predicted by Lewis’ confusability theory to

result in a smaller decrease in acceptability than when case marking is added to the final NP in unscrambled sentences. That is, the unscrambled order NP1-*ne* NP2-**ko** NP3 is predicted to be more acceptable than the scrambled order NP2-**ko** NP1-*ne* NP3, and the reduction in acceptability when an NP sequence like NP1-*ne* NP2-**ko** NP3-**ko** is scrambled to NP2-**ko** NP1-*ne* NP3-**ko** should be smaller than the case where a sequence like NP1-*ne* NP2-**ko** NP3 is scrambled to NP2-**ko** NP1-*ne* NP3. This is because the confusability theory predicts that in a sequence like NP2-**ko** NP1-*ne* NP3-**ko** there will be less retrieval difficulty at a verb since the two -*ko* marked NPs are no longer adjacent and are at the two ends of the list of NPs (as discussed in Section 3.5, the two ends of the NP-list are the indexing positions for recalling items in a list).

4.3.3 Results

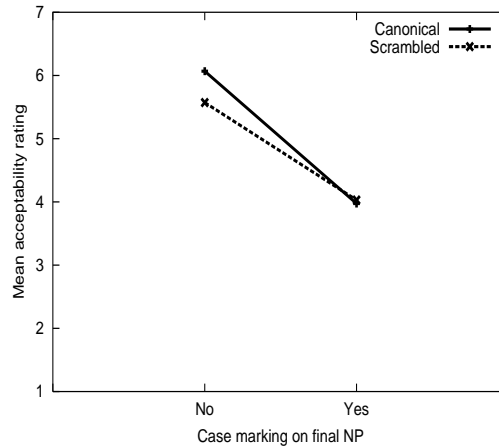


Figure 12: Experiment 3 results

Results showed that items with two -*ko* marked NPs were less acceptable (replicating findings in Experiment 1). Furthermore, as predicted by Lewis' model, scrambling sentences with two -*ko* marked NPs resulted in a smaller decrease in acceptability than scrambling sentences with only one -*ko* marked NP; i.e., there was an interaction between the factors ($F(1,66) = 7.5$).

4.4 Discussion

Consistent with Lewis' theory of positional confusability, reducing similarity of adjacent NPs resulted in a smaller decrease in acceptability. Thus, the data suggest that Lewis' ICT is completely able to account for the retrieval-related processing facts for Hindi, and that the two key components in the ICT play a role in accounting for the data.

5 Conclusion

I empirically evaluated three sentence processing models and showed that Lewis' model makes the best predictions for the Hindi data. I also show that Lewis' model appears to be unable to predict all the processing facts in Hindi. In other work (Vasishth & Kruijff 2000), I propose a model of encoding that can be incorporated into Lewis' sentence processing theory.

An important point is that although the EPDA model clearly fails for Hindi center embeddings, this is not so clear for Gibson's model. Recall that all the sentences in all the experiments were presented out of context, and since we were manipulating specificity of the NP, it is possible that subjects were unable to "accommodate" the specific referent. If this was indeed the source of processing difficulty, then the SPLT's predictions may turn out to be correct if the sentences are presented with appropriate preceding context. Experiments are currently in progress to determine whether this is the case.

Acknowledgments

This research was funded by the Department of Linguistics and the Department of Computer and Information Science, OSU. The experiments were conducted at Jawaharlal Nehru University, New Delhi, India; I thank Professors Ayesha Kidwai, and R. S. Gupta, and Aarti Venkataraman for logistical help. Without their cooperation, this research would have been very difficult to carry out. I am also grateful to Chris Brew, Michael Dickey, David Dowty, Edward Gibson, Martin Jansche, Aravind Joshi, Edith Kaan, Keith Johnson, Neal Johnson, Brian Joseph, Ruth Kempson, Marcus Kracht, Bob Levine, Richard Lewis, Edson Miyamoto, Mineharu Nakayama, Gerald Penn, Shari Speer, and Hans Uszkoreit. Parts of this paper were presented at the CUNY 2001 sentence processing conference at the University of Pennsylvania, at a talk in the Department of Computational Linguistics and Phonetics, The University of Saarland, Germany, and at the 13th European Summer School for Logic, Language, and Information held at Helsinki, Finland; thanks go to the audiences at these venues for their comments and suggestions. Any errors are my sole responsibility. The author's email address is vasishth@ling.ohio-state.edu.

References

- BABYONYSHEV, MARIA, & EDWARD GIBSON. 1999. The complexity of nested structures in Japanese. *Language* 75(3).423–450.
- BACH, E., C. BROWN, & W. MARSLÉN-WILSON. 1986. Crossed and nested dependencies in German and Dutch: A psycholinguistic study. *Language and Cognitive Processes* 1(4).249–262.

- CAPLAN, D., & G. S. WATERS. 1999. Verbal working memory and sentence comprehension. *Behavioral and Brain Sciences* 22.77–94.
- CHOMSKY, NOAM, & GEORGE MILLER. 1963. Introduction to the formal analysis of natural languages. In *Handbook of Mathematical Psychology*, ed. by R. Duncan Luce, Robert R. Bush, & Eugene Galanter, 269–321. John Wiley.
- COHEN, J. D., B. MACWHINNEY, M. FLATT, & J. PROVOST. 1993. PsyScope: An interactive graphic system for designing and controlling experiments in the psychology laboratory using Macintosh computers. *Behavior Research Methods, Instruments and Computers* 25.257–271.
- FERREIRA, F., & JR. CLIFTON, C. 1986. The independence of syntactic processing. *Journal of Memory and Language* 25.348–368.
- GATHERCOLE, SUSAN E., & ALAN D. BADDELEY. 1993. *Working Memory and Language*. Hove, UK: Lawrence Erlbaum Associates.
- GIBSON, EDWARD. 1998. Linguistic complexity: locality of syntactic dependencies. *Cognition* 68.1–76.
- HENSON, R. N. A. 1999. Coding Position in Short-Term Memory. *International Journal of Psychology* 34(5/6).403–409.
- HOPCROFT, JOHN E., & JEFFREY D. ULLMAN. 1979. *Introduction to Automata Theory, Languages and Computation*. Reading, MA: Addison-Wesley.
- HUMPHREYS, M. S., & G. TEHAN. 1998. Cues and codes in working memory tasks. In *On memory: Evolution, Progress, and Reflections on the 20th Anniversary of the Atkinson-Shiffrin model*, ed. by C. Izawa, 127–151. Mahwah, NJ: Erlbaum.
- HYÖNÄ, JUKKA, & HELI HUIJANEN. 1997. Effects of Case Marking and Word Order on Sentence Parsing in Finnish: An Eye-Fixation Study. *Quarterly Journal of Experimental Psychology* 50A(4).841–858.
- JOSHI, ARAVIND K. 1990. Processing crossed and nested dependencies: An automaton perspective on the psycholinguistic results. *Language and Cognitive Processes* 5(1).1–27.
- JUST, M. A, P. A. CARPENTER, & J. D. WOOLLEY. 1982. Paradigms and processes in reading comprehension. *Journal of Experimental Psychology: General* 111(2).228–238.
- KAAN, EDITH, & NADA VASIĆ. 2000. Cross-serial dependencies in Dutch: Evidence for a discourse-based metric? Poster presented at CUNY Sentence Processing Conference, University of California, San Diego.
- KEPPEL, G., & B. J. UNDERWOOD. 1962. Proactive inhibition in short-term retention of single items. *Journal of Verbal Learning and Verbal Behavior* 1.153–161.

- LEWIS, RICHARD L. 1998. Interference in working memory: Retroactive and proactive interference in parsing. Unpublished manuscript, The Ohio State University.
- , & MINEHARU NAKAYAMA. 1999. Determinants of processing complexity in Japanese embeddings: New theory and data. In *Proceedings of the East Asian Psycholinguistics Workshop*, ed. by Mineharu Nakayama & Charles J. Quinn, Stanford, CA. CSLI.
- MOHANAN, TARA. 1994. *Argument Structure in Hindi*. Stanford: CSLI Publications.
- MÜLLER, G. E., & A. PILZECKER. 1900. *Experimentelle Beiträge zur Lehre vom Gedächtniss (Experimental contributions on the theory of memory)*. Leipzig: J. A. Barth.
- NAKATANI, KENTARO, MARIA BABYONYSHEV, & EDWARD GIBSON. 2000. The complexity of nested structures in Japanese. Poster presented at the CUNY Sentence Processing Conference, University of California, San Diego.
- SCHEEPERS, CHRISTOPH, BARBARA HEMFORTH, & LARS KONIECZNY. 1999. Incremental Processing of German Verb-Final Constructions: Predicting the Verb's Minimum (!) Valency. In *The Second International Conference on Cognitive Science, Japan*.
- TRUESWELL, J. C., M. TANENHAUS, & S. M. GARNSEY. 1994. Semantic influences on parsing: Use of thematic role information in syntactic ambiguity resolution. *Journal of Memory and Language* 33.285–318.
- UEHARA, KEIKO, & DIANNE BRADLEY. 1996. The effect of -ga sequences on processing Japanese multiply center-embedded sentences. In *11th Pacific-Asia Conference on Language, Information, and Computation*.
- VASISHTH, SHRAVAN. in progress. *Working memory in sentence comprehension: An empirical investigation of Hindi center embeddings*. Columbus, OH: Ohio State University dissertation.
- , MARTIN JANSCHKE, & CHRIS BREW. A corpus study of Hindi center embeddings. MS, Ohio State University, in progress.
- , & GEERT-JAN M. KRUIJFF. 2000. Processing as abduction: a sentence processing model. In *Proceedings of the Workshop on Linguistic Theory and Grammar Implementation*, Birmingham University, England. <http://www.cs.bham.ac.uk/~esslli/notes/meurers.html>.
- , & ———. 2001. Sentence processing as abduction + deduction. *Ohio State University Working Papers in Linguistics* 56.183–207.