Implications of a Structured Vectorial Semantic Framework for Incremental Interpretation

Stephen Wu, Andrew Exley, William Schuler
University of Minnesota

1 Introduction

Previous corpus studies have shown that the memory usage of a syntactic processing model based on a simple right-corner grammar transform (Schuler et al. 2008, 2010) corresponds to recent estimates of human short-term memory capacity (Miller 1956, Cowan 2001). This article describes a study of the memory usage of a straightforward extension of this model to incremental *semantic* processing, showing the additional constraints imposed by quantifiers in this model create memory demands similar to those created by center embedding in syntax (Chomsky and Miller 1963, Gibson 1998).

The basic incremental parsing model is derived from a probabilistic context-free grammar (PCFG) model via a model-based right-corner transform (Schuler 2009). The right-corner transform is simply the left-right dual of the left-corner transform of Johnson (1998). It conserves memory in the incremental traversal of a pushdown automaton by composing most constituents within the same pushdown store element, expending store space only for syntactically center-embedded constructions.

This incremental parsing model is extended to interactive interpretation. First, a PCFG parser is augmented with headwords and dependency relations (Charniak 1997, Collins 1997), which are probabilistically generated interactively with syntax. This dependency-augmented parser is then used to generate specific individuals from some world model as referents for each constituent, rather than general headword concepts. Thus, the likelihood probability of an utterance given each individual defines a weighted set or denotation for each constituent. Both the likelihood probabilities and dependency relations of this augmented parser are then encapsulated into vectors and matrices, respectively, and the dimension (or length) of these vectors and matrices is

Dialogue & Discourse, Volume X, Article Y:1–xx, 2010
DOI: nn.nnnn/dd

extended to properly calculate set operations like intersection, complementation and union over these denotations. Finally, these changes to the PCFG parser propagate through the model-based right-corner transform to give a straightforward incremental interpreter.

An incremental interpreter defined in this way can interpret first-order quantifiers as well as simple set operations. Existential quantifiers can be calculated implicitly when a new referent is introduced, requiring no additional machinery. However, non-existential quantifiers require a non-linear threshold operator to be introduced, to count individuals and determine whether a quantifier function has been satisfied. These counts depend on restrictor and scope weights, which are stored in the dimension-extended vectors and matrices described above. But since these vectors must store counts for each quantifier, multiple quantifiers cannot be in-element composed through the right-corner transform, like other constituents. As a result, this interactive interpretation model predicts that quantifier nesting consumes memory in the same way that syntactic center-embedding does.

This prediction is evaluated against the same syntactically-annotated corpus used in the purely syntactic evaluations described earlier, comparing quantifiers that have been syntactically center-embedded with those that have not, to determine whether the predicted additional memory costs for quantifier nesting has any effect on the distribution of quantifiers in syntactic contexts that already demand a large amount of memory. The results are positive and statistically significant, supporting the predictions of the model.

The remainder of this article is organized as follows: Section 2 describes some related work in incremental interpretation; Section 3 gives an introduction to probabilistic parsing, including incremental HHMM parsing; Section 4 defines structured vectorial semantics and integrates this framework to into an incremental HHMM parser; Section 5 further explains and tests the predictions of the model; and Section 6 presents some conclusions that can be drawn from this experiment.

2 Related Work

Early approaches to incremental interpretation (Mellish 1985, Haddock 1989) apply semantic constraints associated with each word in a sentence to progressively winnow the set of individuals that could serve as referents in that sentence. These incrementally constrained referents are then used to guide the syntactic analysis of the sentence, dispreferring analyses with empty interpretations in the current environment or discourse context. Similar approaches were applied to broad-coverage

text processing, querying a large commonsense knowledge base as a world model (Martin and Riesbeck 1986). Later, this idea was extended to pursue multiple interpretations at once by exploiting polynomial structure sharing in a dynamic programming parser (DeVault and Stone 2003, Gorniak and Roy 2004, Aist et al. 2007). The resulting shared interpretation is similar to underspecified semantic representations (Bos 1996). This approach was extended to cover hypothetical referents (DeVault and Stone 2003), domains with continuous relations (Gorniak and Roy 2004), and updates to the shared parser chart (Aist et al. 2007). But none of these attempted to incorporate psycholinguistically-motivated bounds on short-term working memory, as the present study does.

Other systems (Elman 1991, Mayberry and Miikkulainen 2003, Dennis et al. 2009) attempt to model incremental processing gracefully degrading as memory demands increase, but do not attempt to model referential specificity of incremental interpretation, as evidenced in eye-tracking experiments such as those by Tanenhaus et al. (1995).

Despite many shared concepts, the goal in this article is not to demonstrate the performance of a user interface in a limited domain, but to evaluate the psycholinguistic plausibility of the incremental interpreter itself, in this case through claims about the effects of quantification in short-term memory.

3 Background

3.1 Notation

This article will associate variables for syntactic categories c, role or relation labels l, referent entities e or vectors of referent entities $\bar{e}$, trees or subtrees τ, and string yields x with constituents in phrase structure trees, identified using subscripts that describe the path from the root of the tree containing this constituent to the constituent itself. These paths may consist of left branches (indicated by '0's in the path) and right branches (indicated by '1's), concatenated into sequences η (or ι or κ). Thus, if a path η identifies a constituent, that constituent's left child would be identified by $\eta 0$, and that constituent's right child would be identified by $\eta 1$.

The probabilistic parsers defined here will also use an indicator function $[\![\cdot]\!]$ to denote deterministic probabilities: $[\![\phi]\!] = 1$ if ϕ is true, 0 otherwise.

3.2 Parsing

For a phrase structure tree rooted at a constituent of category c_η with yield x_η, the task of parsing will require the calculation of the inside

(substring) probability and Viterbi (best substring) probability. When the domain X of x_η is a subset of the domain C of c_η, these can be calculated using probabilities assigned to grammar rules in a probabilistic context-free grammar (PCFG) model θ_{G}, notated:

$$\mathsf{P}_{\theta_{\mathrm{G}}}(c_\eta \rightarrow c_{\eta 0}\ c_{\eta 1}) = \mathsf{P}_{\theta_{\mathrm{G}}}(c_{\eta 0}\ c_{\eta 1} \mid c_\eta) \tag{1}$$

Any yield x_η can be decomposed into prefix $x_{\eta 0}$ and suffix $x_{\eta 1}$ yields:

$$x_\eta = x_{\eta 0} \circ x_{\eta 1} \tag{2}$$

Therefore, inside likelihood probabilities can be defined, marginalizing over all such decompositions (or with probability 1.0 for preterminals $c_\eta = x_\eta$):

$$\mathsf{P}_{\theta_{\mathrm{Ins(G)}}}(x_\eta \mid c_\eta) = \sum_{x_{\eta 0} c_{\eta 0}, x_{\eta 1} c_{\eta 1}} \mathsf{P}_{\theta_{\mathrm{G}}}(c_\eta \rightarrow c_{\eta 0}\ c_{\eta 1}) \cdot \mathsf{P}_{\theta_{\mathrm{Ins(G)}}}(x_{\eta 0} \mid c_{\eta 0}) \cdot \mathsf{P}_{\theta_{\mathrm{Ins(G)}}}(x_{\eta 1} \mid c_{\eta 1}) \tag{3}$$

and Viterbi scores (the probability of the best tree) can be defined, maximizing over all such decompositions (again, with probability 1.0 when $c_\eta = x_\eta$):

$$\mathsf{P}_{\theta_{\mathrm{Vit(G)}}}(x_\eta \mid c_\eta) = \max_{x_{\eta 0} c_{\eta 0}, x_{\eta 1} c_{\eta 1}} \mathsf{P}_{\theta_{\mathrm{G}}}(c_\eta \rightarrow c_{\eta 0}\ c_{\eta 1}) \cdot \mathsf{P}_{\theta_{\mathrm{Vit(G)}}}(x_{\eta 0} \mid c_{\eta 0}) \cdot \mathsf{P}_{\theta_{\mathrm{Vit(G)}}}(x_{\eta 1} \mid c_{\eta 1}) \tag{4}$$

From these, it is possible to obtain the probability of a sentence x:

$$\mathsf{P}(x) = \sum_{c_\eta} \mathsf{P}_{\theta_{\mathrm{Ins(G)}}}(x \mid c_\eta) \cdot \mathsf{P}(c_\eta) \tag{5}$$

and the most likely tree:

$$\hat{\tau}_\eta = \operatorname*{argmax}_{\tau_\eta} \mathsf{P}_{\theta_{\mathrm{Vit(G)}}}(x \mid c_{\tau_\eta}) \cdot \mathsf{P}(c_{\tau_\eta}) \tag{6}$$

3.3 Incremental Parsing

Note that any prefix of a yield x_η can be rewritten as an 'incomplete yield' consisting of the complete yield lacking some suffix of that yield $x_{\eta\iota}$:

$$x_\eta = (x_\eta - x_{\eta\iota}) \circ x_{\eta\iota} \tag{7}$$

It is therefore possible to decompose any inside likelihood (or Viterbi) probability into two parts: first, an 'incomplete constituent' probability of generating this incomplete yield $x_\eta - x_{\eta\iota}$ along with an *awaited* constituent of category $c_{\eta\iota}$, given an *active* constituent of category c_η; and second, an ordinary inside or Viterbi probability (or 'complete con-

stituent' probability) of generating $x_{\eta\iota}$ given $c_{\eta\iota}$:

$$\mathsf{P}_{\theta_{\text{Ins(G)}}}(x_\eta \,|\, c_\eta) = \sum_{\iota \in 1^+, x_{\eta\iota}, c_{\eta\iota}} \mathsf{P}_{\theta_{\text{IC(G)}}}(x_\eta - x_{\eta\iota},\ c_{\eta\iota} \,|\, c_\eta) \cdot \mathsf{P}_{\theta_{\text{Ins(G)}}}(x_{\eta\iota} \,|\, c_{\eta\iota}) \tag{8}$$

This decomposition gives incomplete constituent probabilities that can then be decomposed into other incomplete constituent probabilities:

$$\begin{aligned}
&\mathsf{P}_{\theta_{\text{IC(G)}}}(x_\eta - x_{\eta\iota 1},\ c_{\eta\iota 1} \,|\, c_\eta) \\
&\quad = \sum_{x_{\eta\iota}, c_{\eta\iota}} \mathsf{P}_{\theta_{\text{IC(G)}}}(x_\eta - x_{\eta\iota},\ c_{\eta\iota} \,|\, c_\eta) \cdot \mathsf{P}_{\theta_{\text{IC(G)}}}(x_{\eta\iota} - x_{\eta\iota 1},\ c_{\eta\iota 1} \,|\, c_{\eta\iota}) && (9) \\
&\quad = \sum_{x_{\eta\iota}, c_{\eta\iota}} \mathsf{P}_{\theta_{\text{IC(G)}}}(x_\eta - x_{\eta\iota},\ c_{\eta\iota} \,|\, c_\eta) \\
&\qquad\qquad \cdot \sum_{x_{\eta\iota 0}, c_{\eta\iota 0}} \mathsf{P}_{\theta_{\text{G}}}(c_{\eta\iota} \rightarrow c_{\eta\iota 0}\ c_{\eta\iota 1}) \cdot \mathsf{P}_{\theta_{\text{Ins(G)}}}(x_{\eta\iota 0} \,|\, c_{\eta\iota 0}) && (10)
\end{aligned}$$

or into the product of a grammar rule probability and an inside probability at the end of a sequence of such decompositions:

$$\mathsf{P}_{\theta_{\text{IC(G)}}}(x_\eta - x_{\eta 1},\ c_{\eta 1} \,|\, c_\eta) = \sum_{x_{\eta 0}, c_{\eta 0}} \mathsf{P}_{\theta_{\text{G}}}(c_\eta \rightarrow c_{\eta 0}\ c_{\eta 1}) \cdot \mathsf{P}_{\theta_{\text{Ins(G)}}}(x_{\eta 0} \,|\, c_{\eta 0}) \tag{11}$$

essentially turning right-expanding sequences of constituents (with subscript addresses ending in '1') into left-expanding sequences of incomplete constituents, which can be composed together as they are recognized incrementally from left to right. The decompositions of Equations 8–11, taken together, are a model-based right-corner transform (Schuler 2009).

Estimating probabilities for the beginning and ending of these left-expanding sequences will require the estimation of expected counts for repeated applications of Equation 8, marginalizing over values of x. Since left children and right children are decomposed differently, the expected counts $\theta_{\text{G-RL}^*,d}$ will use PCFG probabilities $\theta_{\text{G-R},d}$ and $\theta_{\text{G-L},d}$ that are conditioned on whether the expanding constituent is a right or left child, and on the center-embedding depth d of the expanding constituent. The expected counts $\theta_{\text{G-RL}^*,d}$ are of constituent categories $c_{\eta\iota}$ anywhere in the left progeny of a right child of category c_η:

$$\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \overset{0}{\rightarrow} c_{\eta 0}\ \ldots) = \sum_{c_{\eta 1}} \mathsf{P}_{\theta_{\text{G-R},d}}(c_\eta \rightarrow c_{\eta 0}\ c_{\eta 1}) \tag{12}$$

$$\begin{aligned}
\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \overset{k}{\rightarrow} c_{\eta 0^k 0}\ \ldots) &= \sum_{c_{\eta 0^k}} \mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \overset{k-1}{\rightarrow} c_{\eta 0^k}\ \ldots) \\
&\quad \cdot \sum_{c_{\eta 0^k 1}} \mathsf{P}_{\theta_{\text{G-L},d}}(c_{\eta 0^k} \rightarrow c_{\eta 0^k 0}\ c_{\eta 0^k 1}) && (13)
\end{aligned}$$

$$\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \overset{*}{\rightarrow} c_{\eta\iota}\ \ldots) = \sum_{k=0}^{\infty} \mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \overset{k}{\rightarrow} c_{\eta\iota}\ \ldots) \tag{14}$$

$$\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \overset{+}{\rightarrow} c_{\eta\iota}\ \ldots) = \mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \overset{*}{\rightarrow} c_{\eta\iota}\ \ldots) - \mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \overset{0}{\rightarrow} c_{\eta\iota}\ \ldots) \tag{15}$$

In practice the infinite sum is estimated to some constant K using value iteration (Bellman 1957).

Probabilities for recognized sequences of incomplete constituents (Equations 8 through 11), and expected left progeny counts for unrecognized sequences of incomplete constituents (Equations 12 through 15) can be combined in a probabilistic pushdown automaton with a bounded pushdown store (to simulate the bounded working memory store of a human comprehender). This is essentially an extension of a Hierarchical Hidden Markov Model (HHMM) (Murphy and Paskin 2001), which obtains a most likely sequence of hidden store states $\hat{s}_{1..T}^{1..D}$ of some length T and some maximum depth D, given a sequence of observations (e.g. words) $x_{1..T}$:

$$\hat{s}_{1..T}^{1..D} \overset{\text{def}}{=} \underset{s_{1..T}^{1..D}}{\operatorname{argmax}} \prod_{t=1}^{T} \mathsf{P}_{\theta_{\text{A}}}(s_t^{1..D} \mid s_{t-1}^{1..D}) \cdot \mathsf{P}_{\theta_{\text{B}}}(x_t \mid s_t^{1..D}) \tag{16}$$

The model generates each successive store only after considering whether each nested sequence of incomplete constituents has completed and reduced:

$$\mathsf{P}_{\theta_{\text{A}}}(s_t^{1..D} \mid s_{t-1}^{1..D}) \overset{\text{def}}{=} \sum_{r_t^1..r_t^D} \prod_{d=1}^{D} \mathsf{P}_{\theta_{\text{R}}}(r_t^d \mid r_t^{d+1} s_{t-1}^d s_{t-1}^{d-1}) \cdot \mathsf{P}_{\theta_{\text{S}}}(s_t^d \mid r_t^{d+1} r_t^d\, s_{t-1}^d s_t^{d-1}) \tag{17}$$

Each store element must therefore contain the active and awaited ($c_{s_t^d}^{\text{A}}$ and $c_{s_t^d}^{\text{W}}$) constituent categories necessary to compute an incomplete constituent probability:

$$s_t^d \overset{\text{def}}{=} \langle c_{s_t^d}^{\text{A}}, c_{s_t^d}^{\text{W}} \rangle \tag{18}$$

and each reduction state must contain the complete constituent category $c_{r_t^d}$ necessary to compute an inside likelihood probability, as well as a flag $f_{r_t^d}$ indicating whether a reduction has taken place (to end a sequence of incomplete constituents):

$$r_t^d \overset{\text{def}}{=} \langle c_{r_t^d}, f_{r_t^d} \rangle \tag{19}$$

The model probabilities for these store elements and reduction states can then be defined (from Murphy and Paskin 2001) to expand a new incomplete constituent after a reduction has taken place ($f_{r_t^d} = 1$), tran-

sition along a sequence of store elements if no reduction has taken place ($f_{r_t^d}=0$), and possibly reduce a store element (terminate a sequence) if the store state below it has reduced ($f_{r_t^{d+1}}=1$):[1]

$$\mathsf{P}_{\theta_{\mathrm{S}}}(s_t^d \mid r_t^{d+1} r_t^d \; s_{t-1}^d s_t^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } f_{r_t^{d+1}}=1,\ f_{r_t^d}=1 : \mathsf{P}_{\theta_{\text{S-Exp},d}}(s_t^d \mid s_t^{d-1}) \\ \text{if } f_{r_t^{d+1}}=1,\ f_{r_t^d}=0 : \mathsf{P}_{\theta_{\text{S-Trn},d}}(s_t^d \mid r_t^{d+1} r_t^d \; s_{t-1}^d s_t^{d-1}) \\ \text{if } f_{r_t^{d+1}}=0,\ f_{r_t^d}=0 : [\![s_t^d = s_{t-1}^d]\!] \end{cases} \tag{20}$$

$$\mathsf{P}_{\theta_{\mathrm{R}}}(r_t^d \mid r_t^{d+1} s_{t-1}^d s_{t-1}^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } f_{r_t^{d+1}}=0 : [\![r_t^d = \mathbf{r}_\perp]\!] \\ \text{if } f_{r_t^{d+1}}=1 : \mathsf{P}_{\theta_{\text{R-Rdn},d}}(r_t^d \mid r_t^{d+1} \; s_{t-1}^d \; s_{t-1}^{d-1}) \end{cases} \tag{21}$$

where $s_t^0 = \mathbf{s}_\top$ and $r_t^{D+1} = \mathbf{r}_\top$ for constants $\mathbf{s}_\top$ (an incomplete root constituent), $\mathbf{r}_\top$ (a complete lexical constituent) and $\mathbf{r}_\perp$ (a null state resulting from the failure of an incomplete constituent to complete). These pushdown automaton operations are then refined for right-corner parsing (from Schuler 2009), distinguishing active transitions $\theta_{\text{G-ACT},d}$ (in which an incomplete constituent is completed, but not reduced, and then immediately expanded to a new incomplete constituent in the same store element) from awaited transitions $\theta_{\text{G-AWT},d}$ (which involve no completion):

$$\mathsf{P}_{\theta_{\text{S-Exp},d}}(s_t^d \mid s_t^{d-1}) \stackrel{\text{def}}{=} \mathsf{P}_{\theta_{\text{G-CEE},d}}(s_t^d \mid s_t^{d-1}) \tag{22}$$

$$\mathsf{P}_{\theta_{\text{S-Trn},d}}(s_t^d \mid r_t^{d+1} r_t^d \; s_{t-1}^d s_t^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } r_t^d = \mathbf{r}_\perp : \mathsf{P}_{\theta_{\text{G-AWT},d}}(s_t^d \mid s_{t-1}^d r_t^{d+1}) \\ \text{if } r_t^d \neq \mathbf{r}_\perp : \mathsf{P}_{\theta_{\text{G-ACT},d}}(s_t^d \mid s_t^{d-1} r_t^d) \end{cases} \tag{23}$$

$$\mathsf{P}_{\theta_{\text{R-Rdn},d}}(r_t^d \mid r_t^{d+1} s_{t-1}^d s_{t-1}^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } c_{r_t^{d+1}} = x_t : \mathsf{P}_{\theta_{\text{G-CER},d}}(r_t^d \mid s_{t-1}^d s_{t-1}^{d-1}) \\ \text{if } c_{r_t^{d+1}} \neq x_t : [\![r_t^d = \mathbf{r}_\perp]\!] \end{cases} \tag{24}$$

These right-corner parsing operations then construct a full inside probability decompositions, using Equations 8 through 11 and Equations 12 through 15, marginalizing out all constituents that are not required in each term:

- for cross-element expansions (CEE):

$$\mathsf{P}_{\theta_{\text{G-CEE},d}}(\langle c_{\eta\iota}, c'_{\eta\iota} \rangle \mid \langle -, c_\eta \rangle) \stackrel{\text{def}}{=} \mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \stackrel{*}{\rightarrow} c_{\eta\iota} \; \ldots) \cdot [\![x_{\eta\iota} = c'_{\eta\iota} = c_{\eta\iota}]\!] \tag{25}$$

- for awaited transitions (AWT), from Equation 10:

$$\mathsf{P}_{\theta_{\text{G-AWT},d}}(\langle c_\eta, c_{\eta\iota 1} \rangle \mid \langle c'_\eta, c_{\eta\iota} \rangle \; c_{\eta\iota 0}) \stackrel{\text{def}}{=} [\![c_\eta = c'_\eta]\!] \cdot \frac{\mathsf{P}_{\theta_{\text{G-R},d}}(c_{\eta\iota} \rightarrow c_{\eta\iota 0} \; c_{\eta\iota 1})}{\sum_{c_{\eta\iota 1}} \mathsf{P}_{\theta_{\text{G-R},d}}(c_{\eta\iota} \rightarrow c_{\eta\iota 0} \; c_{\eta\iota 1})} \tag{26}$$

[1] Here, $[\![\cdot]\!]$ is an indicator function: $[\![\phi]\!] = 1$ if ϕ is true, 0 otherwise.

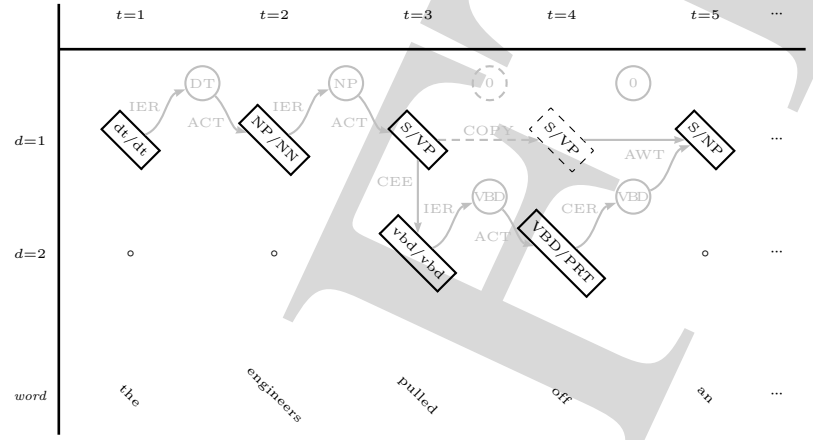

FIGURE 1 Sequence of HHMM store states for a sample sentence "The engineers pulled off an engineering trick," showing a good syntactic hypothesis out of many kept in parallel. Variables corresponding to s_t^d are boxed; r_t^d are circled. Right-corner parsing operations are labeled on the arrows.

$$= [\![c_\eta = c'_\eta]\!] \cdot \frac{\mathsf{P}_{\theta_{\text{G-R},d}}(c_{\eta\iota} \rightarrow c_{\eta\iota 0}\ c_{\eta\iota 1})}{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_{\eta\iota} \overset{0}{\rightarrow} c_{\eta\iota 0}\ \ldots)} \tag{27}$$

- for active transitions (ACT), from Equations 8 and 11:

$$\mathsf{P}_{\theta_{\text{G-ACT},d}}(\langle c_{\eta\iota}, c_{\eta\iota 1}\rangle \,|\, \langle -, c_\eta\rangle\ c_{\eta\iota 0})$$

$$\overset{\text{def}}{=} \frac{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \overset{*}{\rightarrow} c_{\eta\iota}\ \ldots) \cdot \mathsf{P}_{\theta_{\text{G-L},d}}(c_{\eta\iota} \rightarrow c_{\eta\iota 0}\ c_{\eta\iota 1})}{\sum_{c_{\eta\iota}, c_{\eta\iota 1}} \mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \overset{*}{\rightarrow} c_{\eta\iota}\ \ldots) \cdot \mathsf{P}_{\theta_{\text{G}}}(c_{\eta\iota} \rightarrow c_{\eta\iota 0}\ c_{\eta\iota 1})} \tag{28}$$

$$= \frac{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \overset{*}{\rightarrow} c_{\eta\iota}\ \ldots) \cdot \mathsf{P}_{\theta_{\text{G-L},d}}(c_{\eta\iota} \rightarrow c_{\eta\iota 0}\ c_{\eta\iota 1})}{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \overset{+}{\rightarrow} c_{\eta\iota 0}\ \ldots)} \tag{29}$$

- for cross-element reductions (CER):

$$\mathsf{P}_{\theta_{\text{G-CER},d}}(c_{\eta\iota}, \mathbf{1} \,|\, \langle -, c_\eta\rangle\ \langle c'_{\eta\iota}, -\rangle) \overset{\text{def}}{=} [\![c_{\eta\iota} = c'_{\eta\iota}]\!] \cdot \frac{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \overset{0}{\rightarrow} c_{\eta\iota}\ \ldots)}{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \overset{*}{\rightarrow} c_{\eta\iota}\ \ldots)} \tag{30}$$

- for in-element reductions (IER):

$$\mathsf{P}_{\theta_{\text{G-CER},d}}(c_{\eta\iota}, \mathbf{0} \,|\, \langle -, c_\eta\rangle\ \langle c'_{\eta\iota}, -\rangle) \overset{\text{def}}{=} [\![c_{\eta\iota} = c'_{\eta\iota}]\!] \cdot \frac{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \overset{+}{\rightarrow} c_{\eta\iota}\ \ldots)}{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta \overset{*}{\rightarrow} c_{\eta\iota}\ \ldots)} \tag{31}$$

An example analysis is shown in Figure 1.

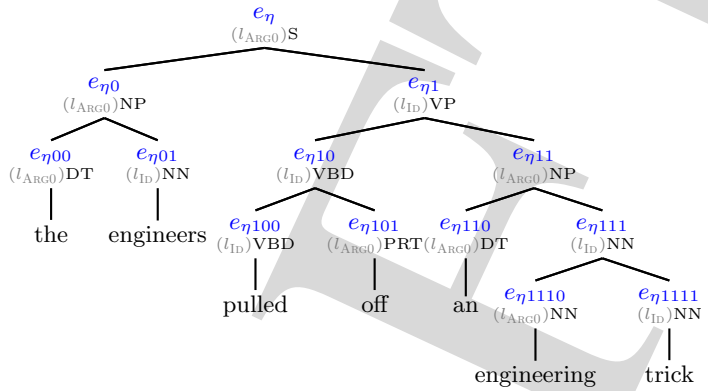

FIGURE 2 Syntax and semantics annotated on a tree. Referents e are subscripted with the node's address. Relations l and syntactic categories c are specific to the example; the relation ARG0 indicates that the NP is the 0^{th} argument of S, and ID shows that the VP is the main predicate of S.

Using Equations 25 through 31, it can be shown that this HHMM produces the same probabilities as the original (rightward-depth-bounded) PCFG. First, any cross-element reduction to $f_{r_t^d} = \mathbf{1}$ must be followed by an awaited transition. This means that in the product of probability terms for a valid parse, every '0' expected count term in the numerator will be cancelled by an equivalent term in the denominator. Similarly, every in-element reduction to $f_{r_t^d} = \mathbf{0}$ must be followed by an active transition. This means that in the product of probability terms for a valid parse, every '+' expected count term in the numerator will be cancelled by an equivalent term in the denominator. Finally, assuming any valid parse ends at time T with $f_{r_T^1} = \mathbf{1}$, it must be the case that every cross-element expansion is eventually followed by a cross-element reduction, with an equal number of active transitions and in-element reductions intervening. This means that in the product of probability terms for a valid parse, every '*' expected count term in the numerator will be cancelled by an equivalent term in the denominator. The remaining terms will be ordinary PCFG expansions.

4 Incremental Interpretation using Structured Vectorial Semantics

To show bounded memory effects for quantifiers, the basic syntactic recognizer described in the previous section is extended to model semantic dependencies and denotations.

4.1 Semantic Dependencies

Semantic referents and relations are tied to grammatical structure. The task of interpretation, then, is to *jointly* determine which grammatical structures and semantic representations best match the observed input. Here, we will define vectorial semantics and revise the parsing of Section 3.2 to incorporate these semantics.

Figure 2 shows an example of a tree that has referents e, relations l, and syntactic constituents c annotated. However, relations are merely helping operators that cause referents to be composed correctly, but would not be part of the syntactic or semantic output of an observed utterance. Therefore, a probabilistic grammar G producing Figure 2 could have a rule with only syntactic categories c and referent entities e, for example:

$$\mathrm{S}{:}e_\eta \rightarrow \mathrm{NP}{:}e_{\eta 0} \;\; \mathrm{VP}{:}e_{\eta 1}$$

with probability according to a grammar model, θ_{G}, for each referent e_η.

Instead of directly generating joint c and e probabilities on the right-hand side above, such a syntactic–semantic grammar rule can be factored further. We employ an independence assumption, that semantic referents are generated before syntactic categories (even if those semantics are underspecified), to obtain:

$$\begin{aligned}\mathsf{P}_{\theta_{\mathrm{G}}}(c_\eta e_\eta \rightarrow c_{\eta 0} e_{\eta 0} \; c_{\eta 1} e_{\eta 1}) \stackrel{\text{def}}{=} \\ \sum_{l_{\eta 0}, l_{\eta 1}} \mathsf{P}_{\theta_{\mathrm{M}}}(c_\eta e_\eta \rightarrow l_{\eta 0} c_{\eta 0} \; l_{\eta 1} c_{\eta 1}) \\ \cdot \mathsf{P}_{\theta_{\mathrm{L}}}(e_{\eta 0} \,|\, l_{\eta 0} \; e_\eta) \cdot \mathsf{P}_{\theta_{\mathrm{L}}}(e_{\eta 1} \,|\, l_{\eta 1} \; e_\eta)\end{aligned} \quad (32)$$

In the θ_{M} model, referents constrain the generation of syntax and of acceptable relations for a child constituent. In the θ_{L} models, referents are probabilistically connected to parent referents on the basis of the θ_{M}-generated child relations. Then, example rules that would take the place of joint c and e dependencies might take the forms:

$$\mathrm{S}{:}e_\eta \rightarrow (l_{\mathrm{ARG0}})\mathrm{NP} \;\; (l_{\mathrm{ID}})\mathrm{VP}$$

where $l_{\mathrm{ARG0}}(e_\eta) = e_{\eta 0}$ and $l_{\mathrm{ID}}(e_\eta) = e_{\eta 1}$, with probabilities according to θ_{M} and θ_{L}, respectively.

4.2 Vectors, Probabilities, and Denotations

Semantic referents can be extended from headword concepts to individuals in some world model. Inside probabilities (probabilities of observed words given a referent and syntactic category) then provide a natural representation for denotations, since individuals are not likely to generate descriptions they do not satisfy.

But in order to define some logical operations on these denotations, it will be necessary to set them apart from the usual purely probabilistic operations of a time-series recognizer; this is done by encapsulating them in vectors and matrices. This encapsulation preserves the probabilities of ordinary dependency relations θ_{L} and expansion rule generation θ_{M}, but can also contain additional elements required to perform non-probabilistic negation, disjunction and quantification operations.

This article will use a double bar to indicate matrices (e.g., relations $\bar{\bar{l}}$), a single bar to indicate vectors (e.g., referents $\bar{e}$), and no bar to indicate any single-valued variable (e.g. labels l that indicate which $\bar{\bar{l}}$ to use). Often, these variables will technically be functions with arguments written in parentheses, producing the expected vectors or matrices (e.g., $\bar{\bar{l}}(l_\eta)$ will produce some matrix based on the argument l_η). Also, indices of vectors and matrices are constants (e.g., $\mathrm{e}_1, \ldots, \mathrm{e}_{|\bar{e}|}$), variables over those indices are single-value variables (e.g., e_η). The contents of vectors and matrices can also be accessed by index (e.g., $\bar{e}[\mathrm{e}_1]$ for a constant, $\bar{e}[e]$ for a variable).

When matrices correspond to relations between constituents, they will be notated with subscripts indicating the addresses of the two constituents related by that matrix: e.g. $\bar{\bar{e}}_{\eta\times\iota}$. This is done in order to show how matrices can be chained together in the traversal of a tree.

Again, vectors and matrices will be used to encapsulate probability distributions. This article will adopt the convention that column components of matrices or vectors represent values of conditioned variables, and row components represent values of modeled variables. Thus, likelihood functions $\mathsf{P}(x \,|\, e)$ are naturally represented as column vectors $\bar{e}$, prior distributions $\mathsf{P}(e)$ are naturally represented as row (or transpose) vectors $\bar{a}^T$, and posterior probabilities $\mathsf{P}(x)$ are vector products[2] $\bar{a}^T \cdot \bar{e}$.

$$\bar{a}^T \cdot \bar{e} = \begin{bmatrix} \mathsf{P}(\mathrm{e}_1) & \mathsf{P}(\mathrm{e}_2) & \cdots \end{bmatrix} \cdot \begin{bmatrix} \mathsf{P}(x \,|\, \mathrm{e}_1) \\ \mathsf{P}(x \,|\, \mathrm{e}_2) \\ \vdots \end{bmatrix} \tag{33}$$

$$= \sum_i \mathsf{P}(\mathrm{e}_i) \cdot \mathsf{P}(x \,|\, \mathrm{e}_i) \tag{34}$$

The indices of $\bar{a}^T$ are written as e_i because the codomain (column indices) of $\bar{a}^T$ must be the same as the domain (row indices) of $\bar{e}$.

Probability-encapsulating vectors can themselves be incorporated

[2] A basic linear algebra operation, the *inner product* (or dot product) of two $m \times 1$ vectors $\bar{u}$ and $\bar{v}$ is written $\bar{u}^\intercal \cdot \bar{v}$ and results in a scalar value $\sum_i u[i] \cdot v[i]$. More generally, the inner product of a $m \times q$ matrix $\bar{\bar{x}}$ and a $q \times n$ matrix $\bar{\bar{y}}$ is a $m \times n$ matrix where $z[i,j] = \sum_k x[i,k] \cdot y[k,j]$.

into other probability distributions. Distributions modeling these vectors and matrices are deterministic, equal to one if the vector or matrix contains the correct probabilities, zero otherwise. Typically, these distributions will involve products of probability terms, and some amount of marginalization over the result. For example, the values of the conditioned vectors or matrices may be marginalized as $\bar{a}_\eta^T$ is, below:

$$\mathsf{P}(\bar{a}_\iota^T \,|\, \bar{a}_\eta^T) \stackrel{\text{def}}{=} [\![\bar{a}_\iota^T = e_\iota \mapsto \sum_{e_\eta} \bar{a}_\eta^T[e_\eta] \cdot \mathsf{P}(e_\iota \,|\, e_\eta)]\!] \tag{35}$$

$$= [\![\bar{a}_\iota^T = \bar{a}_\eta^T \cdot \bar{\bar{l}}_{\eta\times\iota}]\!] \tag{36}$$

where $e_\iota \mapsto$ indicates that what follows it is an element-by-element definition of $\bar{e}_\iota$. The first equality assumes we have the model probability $\mathsf{P}(e_\iota \,|\, e_\eta)$, and the second writes this probability as a matrix $\bar{\bar{l}}_{\eta\times\iota}$ with elements giving the relevant probabilities.

Alternatively, the values in the modeled vectors or matrices may be marginalized as $\bar{e}_\iota$ is, below:

$$\mathsf{P}(\bar{e}_\iota \,|\, \bar{e}_\eta) \stackrel{\text{def}}{=} [\![\bar{e}_\eta = e_\eta \mapsto \sum_{e_\iota} \bar{e}_\iota[e_\iota] \cdot \mathsf{P}(e_\iota \,|\, e_\eta)]\!] \tag{37}$$

$$= [\![\bar{e}_\eta = \bar{\bar{l}}_{\eta\times\iota} \cdot \bar{e}_\iota]\!] \tag{38}$$

Both the probability notation and the linear algebra notation will be used to define a structured vectorial semantics. While using vectors and matrices, the need will occasionally arise to re-weight probabilities without changing indices. This is accomplished by an operation $\mathrm{d}(\bar{e})$ that lists the elements of a column vector on the diagonal of a diagonal matrix. As an example:

$$\bar{e}_\eta = \begin{matrix} \\ \textit{domain} \\ \begin{matrix} \mathrm{e}_4 \\ \mathrm{e}_3 \\ \mathrm{e}_2 \\ \mathrm{e}_1 \end{matrix} \end{matrix} \begin{matrix} \textit{truth} \\ \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} \end{matrix} \qquad \mathrm{d}(\bar{e}_\eta) = \begin{matrix} \\ \textit{domain} \\ \begin{matrix} \mathrm{e}_4 \\ \mathrm{e}_3 \\ \mathrm{e}_2 \\ \mathrm{e}_1 \end{matrix} \end{matrix} \begin{matrix} \textit{codomain} \\ \begin{matrix} \mathrm{e}_1 & \mathrm{e}_2 & \mathrm{e}_3 & \mathrm{e}_4 \end{matrix} \\ \begin{bmatrix} p_1 & 0 & 0 & 0 \\ 0 & p_2 & 0 & 0 \\ 0 & 0 & p_3 & 0 \\ 0 & 0 & 0 & p_4 \end{bmatrix} \end{matrix} \tag{39}$$

where the codomain of $\mathrm{d}(\bar{e})$ is obviously the same as its domain. Thus, any matrix with compatible domains, multiplied by $\mathrm{d}(\bar{e})$, will result in a reweighting of the elements of that matrix.

As a special case, consider a column vector of all ones, $\bar{\mathbf{1}}$; diagonally-listing this vector results in an identity matrix, $\bar{\bar{\mathbf{I}}}$. For any vector $\bar{e}$, the $\mathrm{d}(\cdot)$ operation and multiplying by $\bar{\mathbf{1}}$ are inverse operations: $\mathrm{d}(\bar{e}) \cdot \bar{\mathbf{1}} = \bar{e}$.

Now, with a linear-algebraic notation for probability, the probability models θ_{M} and θ_{L} from the previous section can each be encapsulated in vectors and matrices. First, the syntactic probabilities are captured in a matrix (a diagonally-listed vector), with a different probability for

each possible referent:

$$\bar{m}(c_\eta \to l_{\eta 0} c_{\eta 0}\ l_{\eta 1} c_{\eta 1}) \equiv e_\eta \mapsto \mathsf{P}_{\theta_{\mathrm{M}}}(c_\eta e_\eta \to l_{\eta 0} c_{\eta 0}\ l_{\eta 1} c_{\eta 1}) \tag{40}$$

$$\bar{\bar{m}}(c_\eta \to l_{\eta 0} c_{\eta 0}\ l_{\eta 1} c_{\eta 1}) = \mathrm{d}(\bar{m}(c_\eta \to l_{\eta 0} c_{\eta 0}\ l_{\eta 1} c_{\eta 1})) \tag{41}$$

These can then be conditioned on the center-embedding depth d of the expanding constituent and whether the expanding constituent is a left or right child: $\bar{\bar{m}}_{\mathrm{L},d}$ and $\bar{\bar{m}}_{\mathrm{R},d}$, using $\theta_{\text{M-L},d}$ and $\theta_{\text{M-R},d}$, respectively, in place of θ_{M} above. Next, relation matrices are functions from referents of some constituent η to referents of a child $\eta\iota$ with some probability:

$$\bar{\bar{l}}(l_{\eta\iota}) = \bar{\bar{l}}_{\eta\times\eta\iota}(l_{\eta\iota}) \equiv e_\eta \mapsto e_{\eta\iota} \mapsto \mathsf{P}_{\theta_{\mathrm{L}}}(e_{\eta\iota} \,|\, l_{\eta\iota}\ e_\eta) \tag{42}$$

With these vector and matrix definitions of the probability distributions, θ_{G} can be re-written in terms of vectors, incorporating the per-element probabilities of Equation 32:

$$\begin{aligned}
&\mathsf{P}_{\theta_{\mathrm{G}}}(c_\eta \bar{e}_\eta \to c_{\eta 0} \bar{e}_{\eta 0}\ c_{\eta 1} \bar{e}_{\eta 1}) \\
&\quad = [\![\bar{e}_\eta = e_\eta \mapsto \sum_{e_{\eta 0}, e_{\eta 1}} \mathsf{P}_{\theta_{\mathrm{G}}}(c_\eta e_\eta \to c_{\eta 0} e_{\eta 0}\ c_{\eta 1} e_{\eta 1}) \cdot \bar{e}_{\eta 0}[e_{\eta 0}] \cdot \bar{e}_{\eta 1}[e_{\eta 1}]]\!]
\end{aligned} \tag{43}$$

$$\begin{aligned}
&= [\![\bar{e}_\eta = e_\eta \mapsto \sum_{l_{\eta 0}, l_{\eta 1}} \mathsf{P}_{\theta_{\mathrm{M}}}(c_\eta e_\eta \to l_{\eta 0} c_{\eta 0}\ l_{\eta 1} c_{\eta 1}) \\
&\qquad\qquad \cdot \sum_{e_{\eta 0}} \mathsf{P}_{\theta_{\mathrm{L}}}(e_{\eta 0} \,|\, l_{\eta 0}\ e_\eta) \cdot \bar{e}_{\eta 0}[e_{\eta 0}] \\
&\qquad\qquad \cdot \sum_{e_{\eta 1}} \mathsf{P}_{\theta_{\mathrm{L}}}(e_{\eta 1} \,|\, l_{\eta 1}\ e_\eta) \cdot \bar{e}_{\eta 1}[e_{\eta 1}]]\!]
\end{aligned} \tag{44}$$

$$\begin{aligned}
&= [\![\bar{e}_\eta = \sum_{l_{\eta 0}, l_{\eta 1}} \bar{\bar{m}}(c_\eta \to l_{\eta 0} c_{\eta 0}\ l_{\eta 1} c_{\eta 1}) \\
&\qquad\qquad \cdot \mathrm{d}(\bar{\bar{l}}_{\eta\times\eta 0}(l_{\eta 0}) \cdot \bar{e}_{\eta 0}) \cdot \mathrm{d}(\bar{\bar{l}}_{\eta\times\eta 1}(l_{\eta 1}) \cdot \bar{e}_{\eta 1}) \cdot \bar{n}_\eta]\!]
\end{aligned} \tag{45}$$

where $\bar{n}_\eta = \bar{\mathbf{1}}$. For diagonal matrices $\bar{\bar{m}}$ and $\mathrm{d}(\bar{\bar{l}} \cdot \bar{e})$, this returns a vector after the series of matrix products, without changing the probabilities. Extensions to other values of $\bar{n}_\eta$ will be considered in Section 4.4.

This vectorial representation can be used to define inside probabilities in a semantic parser:[3]

$$\begin{aligned}
\mathsf{P}_{\theta_{\mathrm{Ins(G)}}}(x_{\tau_\eta} \,|\, c_{\tau_\eta} \bar{e}_{\tau_\eta}) &= \mathsf{P}_{\theta_{\mathrm{G}}}(c_{\tau_\eta} \bar{e}_{\tau_\eta} \to c_{\tau_{\eta 0}} \bar{e}_{\tau_{\eta 0}}\ c_{\tau_{\eta 1}} \bar{e}_{\tau_{\eta 1}}) \\
&\quad \cdot \mathsf{P}_{\theta_{\mathrm{Ins(G)}}}(x_{\tau_{\eta 0}} \,|\, c_{\tau_{\eta 0}} \bar{e}_{\tau_{\eta 0}}) \cdot \mathsf{P}_{\theta_{\mathrm{Ins(G)}}}(x_{\tau_{\eta 1}} \,|\, c_{\tau_{\eta 1}} \bar{e}_{\tau_{\eta 1}})
\end{aligned} \tag{46}$$

Finally, given a prior at the start symbol of the grammar,

$$\mathsf{P}_{\pi_{\mathrm{G}}}(c_\eta\ \bar{a}_\eta^T) = [\![\bar{a}_\eta^T = e_\eta \mapsto \mathsf{P}_{\pi_{\mathrm{G}}}(c_\eta\ e_\eta)]\!] \tag{47}$$

[3] Although $\bar{e}$ may be considered a semantic vector, multiplying in $\bar{\bar{m}}$ causes $\bar{e}$ to carry syntactic probabilities as well; of course, it also generates the referent children via $\mathrm{d}(\bar{\bar{l}} \cdot \bar{e})$.

the posterior probability of a sentence is:

$$\mathsf{P}(x) = \sum_{\tau_\eta} \mathsf{P}_{\pi_{\mathrm{G}}}(c_{\tau_\eta} \bar{a}_{\tau_\eta}^T) \cdot \mathsf{P}_{\theta_{\mathrm{Ins(G)}}}(x \,|\, c_{\tau_\eta} \bar{e}_{\tau_\eta}) \cdot \bar{a}_{\tau_\eta}^T \cdot \bar{e}_{\tau_\eta} \tag{48}$$

Again, Viterbi probabilities are simply inside probabilities, replacing summation with maximization.

The goal here is to define a flexible, learnable semantic model that can be extended to a variety of tasks. As two important cases, Section 4.3 describes how this framework can encode a lexicalized parser, and Section 4.4 describes how it can encode a first-order logic interpreter.

As a psycholinguistic model of language, however, the binary-branching CFG paradigm following Equation 32 ignores the crucial issue of incrementality in interpretation. Therefore, Section 4.5 maps the preceding syntactic–semantic equations into an incremental framework for parsing.

4.3 Bilexical Parsing with Structured Vectorial Semantics

Probabilities corresponding to headwords can be encapsulated in vectors and matrices, effectively allowing for bilexical parsing (Charniak 1997, Collins 1997). This definition of vectors and matrices can be considered a form of degenerate semantics, where referent headwords are concepts, and relations are dependencies between these concepts.

Vectors are indexed by individual lexical items, and express the conditional probability that each lexical item will produce an observed yield in syntactic context. For example, at the level of word generation, the headword vector is 0 everywhere but the lexical item corresponding to the yield:

$$\bar{e}_\eta = e_\eta \mapsto \mathsf{P}_{\theta_{\mathrm{Ins(G)}}}(x_\eta \,|\, c_\eta e_\eta) = \begin{matrix} \\ \textit{headwords} \\ \vdots \\ e_{x_\eta} \\ \vdots \end{matrix} \begin{matrix} \textit{truth} \\ \begin{bmatrix} \vdots \\ 1 \\ \vdots \end{bmatrix} \end{matrix} \tag{49}$$

Relation matrices $\bar{\bar{l}}$ are functions from headwords e_η to child headwords $e_{\eta 0}$ or $e_{\eta 1}$, with some probability. As such, this framework easily captures standard bilexical dependencies at each constituent c_η. Typically, a nonterminal constituent will share its headword with one of its sub-constituents, so this sub-constituent will be associated with the identity matrix $\bar{\bar{l}}_{\mathrm{ID}}$ as its labeled role relation. The other sub-constituent will then be associated with a generalized modifier relation $\bar{\bar{l}}_{\mathrm{ARG0}}$, or one of a set of argument-specific relations. Thus, normally:

$$\bar{e}_\eta = \bar{\bar{m}} \cdot \mathrm{d}(\bar{\bar{l}}_{\mathrm{ARG0}} \cdot \bar{e}_{\eta 0}) \cdot \mathrm{d}(\bar{\bar{l}}_{\mathrm{ID}} \cdot \bar{e}_{\eta 1}) \cdot \bar{\mathbf{1}} \;+\; \bar{\bar{m}} \cdot \mathrm{d}(\bar{\bar{l}}_{\mathrm{ID}} \cdot \bar{e}_{\eta 0}) \cdot \mathrm{d}(\bar{\bar{l}}_{\mathrm{ARG0}} \cdot \bar{e}_{\eta 1}) \cdot \bar{\mathbf{1}} \tag{45′}$$

In some bilexical parsers, the lexical head may be conditioned on

phrasal categories — or other local factors such as whether the child is before or after the head. This can be done in structured vectorial semantics by diversifying the set of labeled relations: e.g., $\overline{\overline{l}}_{\text{ARG0,S,NP}}$.

4.4 Logical Interpretation as Structured Vectorial Semantics

The framework of Section 4.1 and 4.2 can also be used to encode logical, model-theoretic interpretation. In a given world model, we will assume a domain $\mathcal{E}$ of individual entities. Vector indices, then, will exhaustively list these entities. With indices tied to individuals in the world model, vectors define sets of denoted individuals (i.e. functors from individuals to set-membership truth values). For example, with $\mathcal{E} = \{\text{e}_1, \text{e}_2, \text{e}_3, \text{e}_4\}$:

$$\begin{array}{c} \\ \textit{individuals} \\ \begin{matrix}\text{e}_4\\ \text{e}_3\\ \text{e}_2\\ \text{e}_1\end{matrix} \end{array} \begin{array}{c} \textit{truth} \\ \begin{bmatrix}1\\0\\1\\1\end{bmatrix} \end{array} = e \mapsto S(e) = \{\text{e}_1, \text{e}_3, \text{e}_4\} \tag{50}$$

implies that e_2 would be unable to produce the correct yield.

Matrices define relations over individuals (i.e. functors from individuals to individuals), encoding sets of ordered pairs. For example:

$$\begin{array}{c} \\ \textit{domain} \\ \begin{matrix}\text{e}_4\\ \text{e}_3\\ \text{e}_2\\ \text{e}_1\end{matrix} \end{array} \begin{array}{c} \textit{codomain} \\ \begin{matrix}\text{e}_1 & \text{e}_2 & \text{e}_3 & \text{e}_4\end{matrix} \\ \begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ \frac{1}{2} & 0 & \frac{1}{2} & 0\\ 0 & 1 & 0 & 0\end{bmatrix} \end{array} = e \mapsto e' \mapsto R(e,e') = \{\langle \text{e}_2, \text{e}_3\rangle, \langle \text{e}_3, \text{e}_1\rangle, \langle \text{e}_3, \text{e}_3\rangle, \langle \text{e}_4, \text{e}_2\rangle\} \tag{51}$$

When there are multiple elements in the codomain for each element of the domain, as is the case for e_3, probabilities are assigned uniformly. This can be thought of as the probability that a speaker will choose each element as an example or 'guide' referent when generating a description, arbitrarily choosing one element when multiple are being described.

Multiplication defines a composition of these functors:

$$\begin{array}{c} \\ \textit{domain } \alpha \\ \begin{matrix}\text{e}_4\\ \text{e}_3\\ \text{e}_2\\ \text{e}_1\end{matrix} \end{array} \begin{array}{c} \textit{codomain } \beta \\ \begin{matrix}\text{e}_1 & \text{e}_2 & \text{e}_3 & \text{e}_4\end{matrix} \\ \begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ \frac{1}{2} & 0 & \frac{1}{2} & 0\\ 0 & 1 & 0 & 0\end{bmatrix} \end{array} \cdot \begin{array}{c} \textit{codomain } \gamma \\ \textit{(truth)} \\ \textit{domain } \beta \\ \begin{matrix}\text{e}_4\\ \text{e}_3\\ \text{e}_2\\ \text{e}_1\end{matrix} \end{array} \begin{array}{c} \\ \\ \\ \begin{bmatrix}1\\0\\1\\1\end{bmatrix} \end{array} = \begin{array}{c} \textit{codomain } \gamma \\ \textit{(truth)} \\ \textit{domain } \alpha \\ \begin{matrix}\text{e}_4\\ \text{e}_3\\ \text{e}_2\\ \text{e}_1\end{matrix} \end{array} \begin{array}{c} \\ \\ \\ \begin{bmatrix}0\\1\\1\\0\end{bmatrix} \end{array} = \{\text{e}_2, \text{e}_3\} \tag{52}$$

$$(\alpha \to \beta) \circ (\beta \to \gamma) = (\alpha \to \gamma)$$

picking out the individuals from the domain (α) of the first functor

that are related through that functor to individuals in the domain (β) of the second. Thus, individuals e from $R(e, e')$ are retained in the set composition only if e' is found in $S(e')$; so, e_4 is dropped in Equation 52.

In this framework, we can now define the typical operations in propositional logic: conjunction and negation; from these, all other propositional statements can be derived.

Definition (Conjunction)**.** *For two vectors $\bar{e}_\alpha$ and $\bar{e}_\beta$, the conjunction is the dot product*

$$\bar{e}_\alpha \wedge \bar{e}_\beta = \mathrm{d}(\bar{e}_\alpha) \cdot \mathrm{d}(\bar{e}_\beta) \cdot \bar{\mathbf{1}} \tag{53}$$

This is the same as set intersection, where membership (a non-zero value) in both $\bar{e}_\alpha$ and $\bar{e}_\beta$ is requisite for membership in the resulting set. For example, in structured vectorial semantics, two vectors $\bar{e}_{\eta 0} = [0\ 1\ 1\ 1]^T$ and $\bar{e}_{\eta 1} = [1\ 1\ 0\ 1]^T$ could first be multiplied by identity relations $\bar{\bar{l}}_{\mathrm{ID}} = \bar{\bar{\mathbf{I}}}$ before being conjoined:

$$\underset{\mathrm{d}(\bar{\bar{l}}_{\mathrm{CONJ0}} \cdot \bar{e}_{\eta 0})}{\begin{array}{c} \\ \mathrm{e}_1 \\ \mathrm{e}_2 \\ \mathrm{e}_3 \\ \mathrm{e}_4 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} \mathrm{e}_1 & \mathrm{e}_2 & \mathrm{e}_3 & \mathrm{e}_4 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}} \cdot \underset{\mathrm{d}(\bar{\bar{l}}_{\mathrm{CONJ1}} \cdot \bar{e}_{\eta 1})}{\begin{array}{c} \\ \mathrm{e}_1 \\ \mathrm{e}_2 \\ \mathrm{e}_3 \\ \mathrm{e}_4 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} \mathrm{e}_1 & \mathrm{e}_2 & \mathrm{e}_3 & \mathrm{e}_4 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}} \cdot \underset{\bar{\mathbf{1}}}{\begin{array}{c} \\ \mathrm{e}_1 \\ \mathrm{e}_2 \\ \mathrm{e}_3 \\ \mathrm{e}_4 \end{array}\!\!\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}} = \underset{\bar{e}_\eta}{\begin{array}{c} \\ \mathrm{e}_1 \\ \mathrm{e}_2 \\ \mathrm{e}_3 \\ \mathrm{e}_4 \end{array}\!\!\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}} \tag{54}$$

This structure should be recognizable from Equation 45 as the means for generating child referents. The default in structured vectorial semantics, then, is to compose denotations of children via conjunction.

Negation (set complementation) can be performed by adding a 'shadow' element for every entity. We will continue to refer to vectors without changing notation, but vector indices of shadow elements will be denoted as e' and appended at the bottom of matrices and vectors.

Definition (Negation)**.** *Let $\bar{e}_\alpha$ be a vector with shadow elements repeating the original vector. Let $\bar{\bar{l}}_{\mathrm{NEG}}$ be a matrix with block structure*

$$\bar{\bar{l}}_{\mathrm{NEG}} = \left[\begin{array}{c:c} -\bar{\bar{\mathbf{I}}} & \bar{\bar{\mathbf{I}}} \\ \hdashline \bar{\bar{\mathbf{0}}} & \bar{\bar{\mathbf{I}}} \end{array}\right],$$

where each block dimension is the size of $\bar{e}_\alpha$ without shadow elements. The negation of $\bar{e}_\alpha$ is

$$\neg \bar{e}_\alpha = \bar{\bar{l}}_{\mathrm{NEG}} \cdot \bar{e}_\alpha \tag{55}$$

As an example of negation:

$$\begin{array}{c} \\ \mathrm{e}_1 \\ \mathrm{e}_2 \\ \mathrm{e}_3 \\ \mathrm{e}'_1 \\ \mathrm{e}'_2 \\ \mathrm{e}'_3 \end{array}\!\!\begin{array}{c} \begin{array}{cccccc} \mathrm{e}_1 & \mathrm{e}_2 & \mathrm{e}_3 & \mathrm{e}'_1 & \mathrm{e}'_2 & \mathrm{e}'_3 \end{array} \\ \left[\begin{array}{cccccc} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \\ \bar{\bar{l}}_{\textsc{Neg}} \end{array} \cdot \begin{array}{c} \\ \mathrm{e}_1 \\ \mathrm{e}_2 \\ \mathrm{e}_3 \\ \mathrm{e}'_1 \\ \mathrm{e}'_2 \\ \mathrm{e}'_3 \end{array}\!\!\begin{array}{c} \textit{truth} \\ \left[\begin{array}{c} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{array}\right] \\ \bar{e}_\eta \end{array} = \begin{array}{c} \\ \mathrm{e}_1 \\ \mathrm{e}_2 \\ \mathrm{e}_3 \\ \mathrm{e}'_1 \\ \mathrm{e}'_2 \\ \mathrm{e}'_3 \end{array}\!\!\begin{array}{c} \textit{truth} \\ \left[\begin{array}{c} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{array}\right] \\ \neg\bar{e}_\eta \end{array} \tag{56}$$

Up until now, the contents of every vector and matrix have been a valid probability distribution or likelihood function. But $\bar{\bar{l}}_{\textsc{Neg}}$ clearly requires non-probabilities, showing that semantic relations in logical interpretation require a formalism, like linear algebra, that is more flexible than pure probabilities.

To reason in first-order logic rather than just propositional logic, we can define quantifiers by using matrix relations and $\bar{n}$. For existential quantifiers, $\bar{n} = \bar{1}$. But for non-existential quantifiers, we will need a cardinality test to determine whether the correct number of referents is denoted. To that end, we define a thresholder, $\bar{n}_{\textsc{Th}}$, to act on a matrix $\bar{\bar{e}}$, comparing row sums of regular elements to row sums of shadow elements.[4]

$$\bar{\bar{e}} \cdot \bar{n}_{\textsc{Th}} = \bar{e} \quad \text{s.t. } \bar{e}[\mathrm{e}_i] = \begin{cases} \sum_{\mathrm{e}_j} \bar{\bar{e}}[\mathrm{e}_i, \mathrm{e}_j], & \text{if } \sum_{\mathrm{e}_j} \bar{\bar{e}}[\mathrm{e}_i, \mathrm{e}_j] \geq \sum_{\mathrm{e}''_j} \bar{\bar{e}}[\mathrm{e}_i, \mathrm{e}''_j] > 0 \\ 0, & \text{otherwise} \end{cases} \tag{57}$$

Then, we can define universal quantification.

Definition (Universal Quantification within a binary relation)**.** *Let S be a unary predicate represented by vector $\bar{e}_\iota$ with shadow elements equal to 1. Let R be a binary predicate with associated relation matrix $\bar{\bar{l}}_R$, with block-diagonal structure*

$$\bar{\bar{l}}_R = \left[\begin{array}{c:c} \bar{\bar{l}}^+_R & \bar{\bar{0}} \\ \hdashline \bar{\bar{0}} & \bar{\bar{l}}'_R \end{array}\right]$$

where $\bar{\bar{l}}^+_R$ is an unadorned relation matrix, and $\bar{\bar{l}}'_R$ is a shadow-element matrix with average values of each row in $\bar{\bar{l}}^+_R$ stored on the diagonal. Let $\bar{\bar{l}}_{\textsc{All}}$ be a matrix with block structure

[4] Different shadow elements are used for negation and quantification, so the complete representations are actually 3×3 block matrices. The extra blocks are omitted for clarity.

$$\bar{\bar{l}}_{\text{ALL}} = \begin{bmatrix} \bar{\bar{\mathbf{1}}} & \bar{\bar{\mathbf{0}}} \\ \hline \bar{\bar{\mathbf{1}}} & \bar{\bar{\mathbf{0}}} \end{bmatrix}$$

where $\bar{\bar{\mathbf{1}}}$ *is a matrix of all ones. Then quantification over vector* $\bar{e}_\alpha$ *is*

$$\bar{e}_\eta \;=\; e_\eta \mapsto \big(\; \forall e_\iota S(e_\iota) \Rightarrow R(e_\eta, e_\iota) \;\big) \;=\; \mathrm{d}(\bar{\bar{l}}_R \cdot (\bar{\bar{l}}_{\text{ALL}} \cdot \bar{e}_\iota)) \cdot \bar{n}_{\text{TH}} \tag{58}$$

We give an abbreviated example of a universal quantifier:

$$\mathrm{d}\left(\begin{array}{c} \begin{array}{c} \\ \mathrm{e}_1 \\ \mathrm{e}_2 \\ \mathrm{e}_3 \\ \mathrm{e}''_1 \\ \mathrm{e}''_2 \\ \mathrm{e}''_3 \end{array} \overset{\begin{array}{cccccc} \mathrm{e}_1 & \mathrm{e}_2 & \mathrm{e}_3 & \mathrm{e}''_1 & \mathrm{e}''_2 & \mathrm{e}''_3 \end{array}}{\left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]} \\ \bar{\bar{l}}_{\text{CONTAINS}} \end{array} \cdot \left(\begin{array}{c} \begin{array}{c} \\ \mathrm{e}_1 \\ \mathrm{e}_2 \\ \mathrm{e}_3 \\ \mathrm{e}''_1 \\ \mathrm{e}''_2 \\ \mathrm{e}''_3 \end{array} \overset{\begin{array}{cccccc} \mathrm{e}_1 & \mathrm{e}_2 & \mathrm{e}_3 & \mathrm{e}''_1 & \mathrm{e}''_2 & \mathrm{e}''_3 \end{array}}{\left[\begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hdashline 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{array}\right]} \\ \bar{\bar{l}}_{\text{ALL}} \end{array} \cdot \begin{array}{c} \begin{array}{c} \\ \mathrm{e}_1 \\ \mathrm{e}_2 \\ \mathrm{e}_3 \\ \mathrm{e}''_1 \\ \mathrm{e}''_2 \\ \mathrm{e}''_3 \end{array} \overset{\begin{array}{c} \textit{truth} \end{array}}{\left[\begin{array}{c} 1 \\ 0 \\ 1 \\ \hdashline 1 \\ 1 \\ 1 \end{array}\right]} \\ \bar{e}_{\text{FILES}} \end{array} \right) \right) \cdot \bar{n}_{\text{TH}}$$

$$= \begin{array}{c} \begin{array}{c} \\ \mathrm{e}_1 \\ \mathrm{e}_2 \\ \mathrm{e}_3 \\ \mathrm{e}''_1 \\ \mathrm{e}''_2 \\ \mathrm{e}''_3 \end{array} \overset{\begin{array}{cccccc} \mathrm{e}_1 & \mathrm{e}_2 & \mathrm{e}_3 & \mathrm{e}''_1 & \mathrm{e}''_2 & \mathrm{e}''_3 \end{array}}{\left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & \frac{2}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{array}\right]} \end{array} \cdot \bar{n}_{\text{TH}} \;=\; \begin{array}{c} \begin{array}{c} \\ \mathrm{e}_1 \\ \mathrm{e}_2 \\ \mathrm{e}_3 \\ \mathrm{e}''_1 \\ \mathrm{e}''_2 \\ \mathrm{e}''_3 \end{array} \overset{\begin{array}{c} \textit{truth} \end{array}}{\left[\begin{array}{c} 0 \\ 1 \\ 0 \\ \hdashline 1 \\ 1 \\ 1 \end{array}\right]} \\ \bar{e}_{\text{CONTAINS}\cdot\,\text{ALL}\cdot\,\text{FILES}} \end{array} \tag{59}$$

where the first term encodes a predicate relation (e.g. $\bar{\bar{l}}_{\text{CONTAINS}}$), the second term encodes a quantifier (e.g. $\bar{\bar{l}}_{\text{ALL}}$), and the third term encodes a property (e.g. $\bar{e}_{\text{FILES}}$). The operations have selected e_2 as the only individual that contains all (two) files, e_1 and e_3. Note that these types of products only occur when two or more PCFG rules are applied; ignoring the $\bar{\bar{m}}$ and eliding identity matrices for clarity, the above sequence could have originally occurred as (applying Equation 45 at two levels):

$$\bar{e}_\alpha = \bar{\bar{m}} \cdot \mathrm{d}(\bar{\mathbf{1}}) \cdot \mathrm{d}(\bar{\bar{l}}_{\text{CONTAINS}} \cdot \bar{\bar{m}} \cdot \mathrm{d}(\bar{\mathbf{1}}) \cdot \mathrm{d}(\bar{\bar{l}}_{\text{ALL}} \cdot \bar{e}_\alpha) \cdot \bar{\mathbf{1}}) \cdot \bar{n}_{\text{TH}} \tag{60}$$

With these operations, this model is able to perform model-theoretic interpretation in first-order logic.

4.5 Incremental Structured Vectorial Semantics in HHMM Parsing

As HHMM probabilities were defined directly on syntactic CFG rules before, they will now be defined on structured vectorial semantics for CFG rules.

First, the incremental parsing probabilities need to be updated with referents encapsulated probabilities in vectors. These are derived from the right-corner decomposition in Equations 8 through 11, extended to include referents:

$$\mathsf{P}_{\theta_{\text{Ins(G)}}}(x_\eta \,|\, c_\eta, e_\eta) = \sum_{c_{\eta\iota}, e_{\eta\iota}} \mathsf{P}_{\theta_{\text{IC(G)}}}(x_\eta{-}x_{\eta\iota}, c_{\eta\iota}, e_{\eta\iota} \,|\, c_\eta, e_\eta) \cdot \mathsf{P}_{\theta_{\text{Ins(G)}}}(x_{\eta\iota} \,|\, c_{\eta\iota}, e_{\eta\iota}) \quad (61)$$

$$\begin{aligned}\mathsf{P}_{\theta_{\text{IC(G)}}}(x_\eta{-}x_{\eta\iota 1}, c_{\eta\iota 1} e_{\eta\iota 1} \,|\, c_\eta e_\eta) = \\ \sum_{x_{\eta\iota} c_{\eta\iota} e_{\eta\iota}} \mathsf{P}_{\theta_{\text{IC(G)}}}(x_\eta{-}x_{\eta\iota}, c_{\eta\iota} e_{\eta\iota} \,|\, c_\eta e_\eta) \cdot \sum_{l_{\eta\iota 0}, l_{\eta\iota 1}} \mathsf{P}_{\theta_{\text{M}}}(c_{\eta\iota} \rightarrow l_{\eta\iota 0} c_{\eta\iota 0} l_{\eta\iota 1} c_{\eta\iota 1}) \\ \cdot \sum_{e_{\eta\iota 0}} \mathsf{P}_{\theta_{\text{L}}}(e_{\eta\iota 0} \,|\, l_{\eta\iota 0} e_{\eta\iota}) \cdot \mathsf{P}_{\theta_{\text{Ins(G)}}}(x_{\eta\iota 0} \,|\, c_{\eta\iota 0} e_{\eta\iota 0}) \cdot \sum_{e_{\eta\iota 1}} \mathsf{P}_{\theta_{\text{L}}}(e_{\eta\iota 1} \,|\, l_{\eta\iota 1} e_{\eta\iota})\end{aligned} \quad (62)$$

$$\begin{aligned}\mathsf{P}_{\theta_{\text{IC(G)}}}(x_\eta{-}x_{\eta 1}, c_{\eta 1}, e_{\eta 1} \,|\, c_\eta, e_\eta) = \sum_{c_{\eta 0}, l_{\eta 0}, l_{\eta 1}} \mathsf{P}_{\theta_{\text{M}}}(c_\eta e_\eta \rightarrow l_{\eta 0} c_{\eta 0}\ l_{\eta 1} c_{\eta 1}) \\ \cdot \sum_{x_{\eta 0}, e_{\eta 0}} \mathsf{P}_{\theta_{\text{L}}}(e_{\eta 0} \,|\, l_{\eta 0}\ e_\eta) \cdot \mathsf{P}_{\theta_{\text{Ins(G)}}}(x_{\eta 0} \,|\, c_{\eta 0}, e_{\eta 0}) \cdot \mathsf{P}_{\theta_{\text{L}}}(e_{\eta 1} \,|\, l_{\eta 1}\ e_\eta)\end{aligned} \quad (63)$$

The deterministic probability of encapsulating these distributions in vectors is then a straightforward substitution of matrices and vectors for θ_{M}, θ_{L}, and $\theta_{\text{Ins(G)}}$ terms. For example, Equation 62 becomes:

$$\begin{aligned}\mathsf{P}(\langle c_\eta, c_{\eta\iota 1}, \bar{\bar{e}}_{\eta\times\eta\iota 1}\rangle \,|\, \langle c_{\eta\iota 0}, \bar{e}_{\eta\iota 0}\rangle) \ = \ [\![\bar{\bar{e}}_{\eta\times\eta\iota 1} = \bar{\bar{e}}_{\eta\times\eta\iota} \\ \cdot \sum_{l_{\eta\iota 0}, l_{\eta\iota 1}} \bar{\bar{m}}(c_{\eta\iota} \rightarrow l_{\eta\iota 0} c_{\eta\iota 0}\ l_{\eta\iota 1} c_{\eta\iota 1}) \cdot \mathrm{d}(\bar{\bar{l}}_{\eta\iota\times\eta\iota 0}(l_{\eta\iota 0}) \cdot \bar{e}_{\eta\iota 0}) \cdot \bar{\bar{l}}_{\eta\iota\times\eta\iota 1}(l_{\eta\iota 1})]\!]\end{aligned} \quad (64)$$

Now, HHMM reduce and store states can be updated to reflect vectorial semantics. Reduction states must contain the variables necessary to compute inside likelihood probabilities for complete constituents, as well as a flag indicating whether a reduction has occurred. Following assumptions in Section 4.2, since a single entity variable appears in the condition of a likelihood probability, a column vector is used:

$$r_\eta \stackrel{\text{def}}{=} \langle\ c_\eta,\ e_\eta \mapsto \sum_{x_\eta} \mathsf{P}_{\theta_{\text{Ins(G)}}}(x_\eta \,|\, c_\eta, e_\eta),\ f_\eta\ \rangle \quad (65)$$

$$= \langle\ c_\eta,\ \bar{e}_\eta,\ f_\eta\ \rangle \quad (66)$$

Store elements must contain the variables necessary to compute incomplete constituent probabilities. Since entities appear as both conditioned and modeled variables in incomplete constituent probabilities,

a matrix is used:

$$s_{\eta\times\eta\iota} \overset{\text{def}}{=} \langle\ c_\eta,\ c_{\eta\iota},\ e_\eta \mapsto e_{\eta\iota} \mapsto \sum_{x_\eta, x_{\eta\iota}} \mathsf{P}_{\theta_{\text{IC(G)}}}(x_\eta - x_{\eta\iota}, c_{\eta\iota}, e_{\eta\iota} \,|\, c_\eta, e_\eta)\ \rangle \quad (67)$$

$$= \langle\ c_\eta,\ c_{\eta\iota},\ \bar{\bar{e}}_{\eta\times\eta\iota}\ \rangle \quad (68)$$

These HHMM variables can define chains of matrices and vectors that jointly represent syntactic and semantic information.

HHMM probabilities with vectorial semantics are then derived from the analagous syntax-only versions in Equations 25 through 31. Syntactic probabilities $\theta_{\text{M-L},d}$ and $\theta_{\text{M-R},d}$ are substituted with matrices $\bar{\bar{m}}_{\text{L},d}$ and $\bar{\bar{m}}_{\text{R},d}$ according to Equation 41 and semantic relation probabilities θ_{L} are substituted with matrices $\bar{\bar{l}}$ according to Equation 42. Similar to the syntactic version, estimate normalization terms (here, inverted matrices instead of denominators) will cancel out, yielding the same probabilities as the semantics-augmented PCFG.

The complete set of vectorial semantic HHMM probabilities are as follows (the full derivations are in the appendix):

- Cross-element expansion (CEE):

$$\mathsf{P}_{\theta_{\text{G-CEE},d}}(\langle c_{\eta\iota}, c'_{\eta\iota}, \bar{\bar{e}}_{\eta\iota\times\eta\iota}\rangle \,|\, \langle -, c_\eta, -\rangle)$$
$$= [\![x_{\eta\iota} = c'_{\eta\iota} = c_{\eta\iota}]\!] \cdot [\![\bar{\bar{e}}_{\eta\iota\times\eta\iota} = \bar{\bar{l}}^{*}_{\eta\times\eta\iota}(c_\eta, c_{\eta\iota}) \cdot \bar{\bar{\mathbf{I}}}_{\eta\iota\times\eta\iota}]\!] \quad (69)$$

- Awaited transition (AWT):

$$\mathsf{P}_{\theta_{\text{G-AWT},d}}(\langle c_\eta, c_{\eta\iota 1}, \bar{\bar{e}}_{\eta\times\eta\iota 1}\rangle \,|\, \langle c'_\eta, c_{\eta\iota}, \bar{\bar{e}}_{\eta\times\eta\iota}\rangle\ c_{\eta\iota 0}\bar{e}_{\eta\iota 0})$$
$$= [\![c_\eta = c'_\eta]\!] \cdot [\![\bar{\bar{e}}_{\eta\times\eta\iota 1} = \bar{\bar{e}}_{\eta\times\eta\iota} \cdot \textstyle\sum_{l_{\eta\iota 0}, l_{\eta\iota 1}} \bar{\bar{m}}_{\text{R},d}(c_{\eta\iota} \to l_{\eta\iota 0} c_{\eta\iota 0}\ l_{\eta\iota 1} c_{\eta\iota 1})$$
$$\cdot\, \text{d}(\bar{\bar{l}}_{\eta\iota\times\eta\iota 0}(l_{\eta\iota 0}) \cdot \bar{\bar{l}}^{0}_{\eta\iota\times\eta\iota 0}(c_{\eta\iota}, c_{\eta\iota 0})^{-1} \cdot \bar{e}_{\eta\iota 0}) \cdot \bar{\bar{l}}_{\eta\iota\times\eta\iota 1}(l_{\eta\iota 1})]\!] \quad (70)$$

- Active transition (ACT):

$$\mathsf{P}_{\theta_{\text{G-ACT},d}}(\langle c_{\eta\iota}, c_{\eta\iota 1}, \bar{\bar{e}}_{\eta\iota\times\eta\iota 1}\rangle \,|\, \langle -, c_\eta, -\rangle\ c_{\eta\iota 0}\bar{e}_{\eta\iota 0})$$
$$= [\![\bar{\bar{e}}_{\eta\iota\times\eta\iota 1} = \bar{\bar{l}}^{*}_{\eta\times\eta\iota}(c_\eta, c_{\eta\iota}) \cdot \textstyle\sum_{l_{\eta\iota 0}, l_{\eta\iota 1}} \bar{\bar{m}}_{\text{L},d}(c_{\eta\iota} \to l_{\eta\iota 0} c_{\eta\iota 0}\ l_{\eta\iota 1} c_{\eta\iota 1})$$
$$\cdot\, \text{d}(\bar{\bar{l}}_{\eta\iota\times\eta\iota 0}(l_{\eta\iota 0}) \cdot \bar{\bar{l}}^{+}_{\eta\times\eta\iota 0}(c_\eta, c_{\eta\iota 0})^{-1} \cdot \bar{e}_{\eta\iota 0}) \cdot \bar{\bar{l}}_{\eta\iota\times\eta\iota 1}(l_{\eta\iota 1})]\!] \quad (71)$$

- Cross-element reduction (CER):

$$\mathsf{P}_{\theta_{\text{G-CER},d}}(c_{\eta\iota}\bar{e}_{\eta\iota}, \mathbf{1} \,|\, \langle -, c_\eta, -\rangle\ \langle c'_{\eta\iota}, c_{\eta\iota\kappa}, \bar{\bar{e}}_{\eta\iota\times\eta\iota\kappa}\rangle)$$
$$= [\![\bar{e}_{\eta\iota} = \bar{\bar{l}}^{0}_{\eta\times\eta\iota}(c_\eta, c_{\eta\iota}) \cdot \bar{\bar{l}}^{*}_{\eta\times\eta\iota}(c_\eta, c_{\eta\iota})^{-1} \cdot \bar{\bar{e}}'_{\eta\iota\times\eta\iota\kappa} \cdot \bar{n}_{\eta\iota}]\!] \quad (72)$$

- In-element reduction (IER):

$$\mathsf{P}_{\theta_{\text{G-CER},d}}(c_{\eta\iota}\bar{e}_{\eta\iota}, \mathbf{0} \,|\, \langle -, c_\eta, -\rangle\ \langle c'_{\eta\iota}, c_{\eta\iota\kappa}, \bar{\bar{e}}_{\eta\iota\times\eta\iota\kappa}\rangle)$$
$$= [\![\bar{e}_{\eta\iota} = \bar{\bar{l}}^{+}_{\eta\times\eta\iota}(c_\eta, c_{\eta\iota}) \cdot \bar{\bar{l}}^{*}_{\eta\times\eta\iota}(c_\eta, c_{\eta\iota})^{-1} \cdot \bar{\bar{e}}'_{\eta\iota\times\eta\iota\kappa} \cdot \bar{n}_{\eta\iota}]\!] \quad (73)$$

where $\bar{n} = \bar{\mathbf{1}}$ has been used in the derivations.

Expected counts in approximated denotations are estimated using value iteration, analogous to Equations 12 through 15 in Section 3:

$$\bar{\bar{l}}^0(c_\eta, c_{\eta 0}) = \sum_{c_{\eta 1}} \sum_{l_{\eta 0}, l_{\eta 1}} \bar{\bar{m}}_{\mathrm{R},d}(c_\eta \to l_{\eta 0} c_{\eta 0}\ l_{\eta 1} c_{\eta 1}) \cdot \bar{\bar{l}}_{\eta \times \eta 0}(l_{\eta 0}) \tag{74}$$

$$\bar{\bar{l}}^k(c_\eta, c_{\eta 0^k 0}) = \sum_{c_{\eta 0^k}} \bar{\bar{l}}^{k-1}(c_\eta, c_{\eta 0^k}) \cdot \sum_{c_{\eta 0^k 1}} \sum_{l_{\eta 0^k 0}, l_{\eta 0^k 1}} \bar{\bar{m}}_{\mathrm{L},d}(c_{\eta 0^k} \to l_{\eta 0^k 0} c_{\eta 0^k 0}\ l_{\eta 0^k 1} c_{\eta 0^k 1}) \cdot \bar{\bar{l}}_{\eta 0^k \times \eta 0^k 0}(l_{\eta 0^k 0}) \tag{75}$$

$$\bar{\bar{l}}^*(c_\eta, c_{\eta\iota}) = \sum_{k=0}^{\infty} \bar{\bar{l}}^k(c_\eta, c_{\eta\iota}) \approx \sum_{k=0}^{K} \bar{\bar{l}}^k(c_\eta, c_{\eta\iota}) \tag{76}$$

$$\bar{\bar{l}}^+(c_\eta, c_{\eta\iota}) = \bar{\bar{l}}^*(c_\eta, c_{\eta\iota}) - \bar{\bar{l}}^0(c_\eta, c_{\eta\iota}) \tag{77}$$

5 Discussion

5.1 Predictions

The previous section extends a basic head dependency semantics to model-theoretic interpretation, then incrementalizes this model using a simple transform to minimize working memory demands during recognition. This model makes interesting predictions about memory demands of non-existential quantifiers.

Compare a complete constituent vector:

$$\bar{e}_{\eta\iota} = \sum \bar{\bar{m}} \cdot \mathrm{d}(\bar{\bar{l}}_{\eta\iota \times \eta\iota 0} \cdot \bar{e}_{\eta\iota 0}) \cdot \mathrm{d}(\bar{\bar{l}}_{\eta\iota \times \eta\iota 1} \cdot \bar{e}_{\eta\iota 1}) \cdot \bar{n}_{\eta\iota} \tag{45$'$}$$

with an incomplete constituent matrix from an awaited transition:

$$\bar{\bar{e}}_{\eta \times \eta\iota 1} = \bar{\bar{e}}_{\eta \times \eta\iota} \cdot \sum \bar{\bar{m}} \cdot \mathrm{d}(\bar{\bar{l}}_{\eta\iota \times \eta\iota 0} \cdot \bar{e}_{\eta\iota 0}) \cdot\ \bar{\bar{l}}_{\eta\iota \times \eta\iota 1} \tag{64$'$}$$

Of course, the right-tail recursion of Equation 64′ lacks the complete constituent $\bar{e}_{\eta\iota 1}$ (which makes the incomplete constituent incomplete); however, Equation 64′ also lacks $\bar{n}_{\eta\iota}$. This is normally reasonable, because if $\bar{n} = \bar{\mathbf{1}}$ then $\mathrm{d}(\bar{\bar{l}} \cdot \bar{e}) \cdot \bar{n} = \bar{\bar{l}} \cdot \bar{e}$. Thus, whenever the awaited $\bar{e}_{\eta\iota 1}$ is fully recognized, $\bar{n} = \bar{\mathbf{1}}$ is tacitly applied.

However, the assumption $\bar{n} = \bar{\mathbf{1}}$ does not hold in the case of universal quantification, as defined in Section 4.4. Recall that universal quantification required $\bar{n}_{\mathrm{TH}}$ at the end of a parent constituent's operations. This would be ignored when invoking an awaited transition. In fact, any non-existential quantifier would require a similar non-linear operator $\bar{n}$ to carry out the desired cardinality check. Since any $\bar{n} \neq \bar{\mathbf{1}}$ cannot be tacitly applied along with a completed tail $\bar{e}_{\eta\iota 1}$, awaited transitions (which expand without invoking a new store element) are not licensed for non-existential quantifiers.

Determiner	Right-Corner Stack Depth							
	1		2		3		4	
all	446	39.7%	577	51.3%	99	8.8%	2	0.1%
both	277	61.0%	165	36.3%	12	2.6%	0	0.0%
each	187	45.3%	199	48.2%	27	6.5%	0	0.0%
every	58	30.4%	112	58.6%	19	9.9%	2	1.0%
half	18	18.6%	59	60.8%	19	19.6%	1	1.0%
no	179	25.9%	445	64.3%	66	9.5%	2	0.3%
Non-exist.	1165	39.2%	1557	52.4%	242	8.1%	7	0.2%
Other DTs	29398	35.2%	45986	55.1%	7901	9.5%	248	0.3%
All DTs	30563	35.3%	47543	55.0%	8143	9.4%	255	0.3%

TABLE 1 Non-existential quantifiers and other determiners occurring at each stack depth.

With this complexity in non-existential quantification a correct application of this kind of quantification requires another HHMM store element. Thus, we expect to see non-existential quantification occurring more commonly at lower HHMM depths than other, grammatically similar words.

This accords with intuition, since sentences with nested quantifiers may easily seem difficult to process ('did a person drive a car into a lake?' vs. 'did every person drive no car into two lakes?'); but this prediction is crucially based on the mapping of denotations to memory elements described in Section 4.5.

5.2 Corpus Analysis

To test this prediction, occurrences of non-existential quantifiers *all, both, each, every, half,* and *no* were examined in the Penn Treebank Wall Street Journal corpus, sections 2 through 21. These were chosen because they were the most frequent non-existential quantifiers tagged as determiners in the corpus. The trees in this corpus were converted to right-corner form, then analyzed to determine the depth of each determiner when mapped to a bounded store of incomplete constituents.

Table 1 shows that non-existential quantifiers behave differently than other determiners, in that they are more likely to appear in lower store depths. The difference in percentage is highly significant: $p = 3.5 * 10^{-6}$ on a t-test comparing non-existential quantifiers with all determiners. This suggests that non-existential quantifiers may indeed require more memory resources than existential quantifiers or non-quantifiers, as predicted by the model.

Some of the quantifiers appear to resist this trend: for example, *half* appears to be common at depth 3. Investigation into this showed that some sentences had the word *half* incorrectly tagged as a DT. For example,

James E. Ousley, computer products group president, said such arrangements help slash Control Data's computer research and development costs in half by the end of 1990

In this particular sentence, the word shows up in depth 3, a position that the model would predict to be much less likely than lower depths. This sort of tagging error was difficult to programmatically avoid, and may have accounted for some of the outlying data.

6 Conclusion

An incremental account of CFG parsing, derived from first principles, gives rise to incomplete constituent probabilities that are naturally captured in a right-corner grammatical formulation. Therefore, by defining semantics to be integrally tied to CFG structure, this article has provided an integrated model of syntactic–semantic language processing that bears all the marks of psycholinguistically-plausible, bounded-memory, right-corner parsing.

The structured semantics defined were encapsulated in vectors and matrices, allowing for first-order logical interpretation on a world model of individual entities. Because non-existential quantifiers required an additional cardinality test compared to other semantic operations, the incremental semantic model made the prediction that these particular quantifiers would occupy an element of short-term memory that would otherwise be used for syntactic processing. A corpus study of the distribution of quantifiers in modeled short-term memory validated this prediction.

Incremental structured vectorial semantics has been defined for the operations of a pushdown automaton with a limited pushdown store, and it is therefore amenable to practical implementation in an HHMM. It can thus be envisioned that headwords, model-theoretic logic, or many other definitions of vector semantics may be incrementally parsed in this standard framework.

Appendix A: Derivation of HHMM Probabilities under Structured Vectorial Semantics

- Cross-element expansion (CEE):

$$\mathsf{P}_{\theta_{\text{G-CEE},d}}(\langle c_{\eta\iota}, c'_{\eta\iota}, \bar{\bar{e}}_{\eta\iota\times\eta\iota}\rangle \,|\, \langle -, c_\eta, -\rangle) \tag{A.1}$$

$$\stackrel{\text{def}}{=} [\![x_{\eta\iota} = c'_{\eta\iota} = c_{\eta\iota}]\!] \cdot [\![\bar{\bar{e}}_{\eta\iota\times\eta\iota} = e_\eta \mapsto e'_{\eta\iota} \mapsto \sum_{e_{\eta\iota}} \mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta e_\eta \stackrel{*}{\rightarrow} c_{\eta\iota} e_{\eta\iota} \ldots) \cdot [\![e'_{\eta\iota} = e_{\eta\iota}]\!]]\!] \tag{A.2}$$

$$= [\![x_{\eta\iota} = c'_{\eta\iota} = c_{\eta\iota}]\!] \cdot [\![\bar{\bar{e}}_{\eta\iota\times\eta\iota} = \bar{\bar{l}}^{*}_{\eta\times\eta\iota}(c_\eta, c_{\eta\iota}) \cdot \bar{\bar{\mathbf{I}}}_{\eta\iota\times\eta\iota}]\!] \tag{A.3}$$

- Awaited transition (AWT):

$$\begin{aligned}
&\mathsf{P}_{\theta_{\text{G-AWT},d}}(\langle c_\eta, c_{\eta\iota 1}, \bar{\bar{e}}_{\eta\times\eta\iota 1}\rangle \,|\, \langle c'_\eta, c_{\eta\iota}, \bar{\bar{e}}_{\eta\times\eta\iota}\rangle \; c_{\eta\iota 0}\bar{e}_{\eta\iota 0}) \\
&\overset{\text{def}}{=} [\![c_\eta = c'_\eta]\!] \cdot [\![\bar{\bar{e}}_{\eta\times\eta\iota 1} = e_\eta \mapsto e_{\eta\iota 1} \mapsto \textstyle\sum_{e_{\eta\iota}, e_{\eta\iota 0}} \bar{\bar{e}}_{\eta\times\eta\iota}[e_\eta, e_{\eta\iota}] \cdot \bar{e}_{\eta\iota 0}[e_{\eta\iota 0}] \\
&\quad \cdot \frac{\mathsf{P}_{\theta_{\text{G-R},d}}(c_{\eta\iota}e_{\eta\iota} \to c_{\eta\iota 0}e_{\eta\iota 0} \;\; c_{\eta\iota 1}e_{\eta\iota 1})}{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_{\eta\iota}e_{\eta\iota} \overset{0}{\to} c_{\eta\iota 0}e_{\eta\iota 0} \;\; \ldots)}]\!]
\end{aligned} \tag{A.4}$$

$$\begin{aligned}
&= [\![c_\eta = c'_\eta]\!] \cdot [\![\bar{\bar{e}}_{\eta\times\eta\iota 1} = e_\eta \mapsto e_{\eta\iota 1} \mapsto \textstyle\sum_{e_{\eta\iota}} \bar{\bar{e}}_{\eta\times\eta\iota}[e_\eta, e_{\eta\iota}] \\
&\quad \cdot \textstyle\sum_{l_{\eta\iota 0}, l_{\eta\iota 1}} \mathsf{P}_{\theta_{\text{M-R},d}}(c_{\eta\iota}e_{\eta\iota} \to l_{\eta\iota 0}c_{\eta\iota 0} \;\; l_{\eta\iota 1}c_{\eta\iota 1}) \\
&\quad \cdot \textstyle\sum_{e_{\eta\iota 0}} \mathsf{P}_{\theta_{\text{L}}}(e_{\eta\iota 0} \,|\, l_{\eta\iota 0} \; e_{\eta\iota}) \cdot \mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_{\eta\iota}e_{\eta\iota} \overset{0}{\to} c_{\eta\iota 0}e_{\eta\iota 0} \; \ldots)^{-1} \\
&\quad \cdot \bar{e}_{\eta\iota 0}[e_{\eta\iota 0}] \cdot \mathsf{P}_{\theta_{\text{L}}}(e_{\eta\iota 1} \,|\, l_{\eta\iota 1} \; e_{\eta\iota})]\!]
\end{aligned} \tag{A.5}$$

$$\begin{aligned}
&= [\![c_\eta = c'_\eta]\!] \cdot [\![\bar{\bar{e}}_{\eta\times\eta\iota 1} = \bar{\bar{e}}_{\eta\times\eta\iota} \cdot \textstyle\sum_{l_{\eta\iota 0}, l_{\eta\iota 1}} \bar{\bar{m}}_{\text{R},d}(c_{\eta\iota} \to l_{\eta\iota 0}c_{\eta\iota 0} \; l_{\eta\iota 1}c_{\eta\iota 1}) \\
&\quad \cdot \mathrm{d}(\bar{\bar{l}}_{\eta\iota\times\eta\iota 0}(l_{\eta\iota 0}) \cdot \bar{\bar{l}}^{0}_{\eta\iota\times\eta\iota 0}(c_{\eta\iota}, c_{\eta\iota 0})^{-1} \cdot \bar{e}_{\eta\iota 0}) \cdot \bar{\bar{l}}_{\eta\iota\times\eta\iota 1}(l_{\eta\iota 1})]\!]
\end{aligned} \tag{A.6}$$

- Active transition (ACT):

$$\begin{aligned}
&\mathsf{P}_{\theta_{\text{G-ACT},d}}(\langle c_{\eta\iota}, c_{\eta\iota 1}, \bar{\bar{e}}_{\eta\iota\times\eta\iota 1}\rangle \,|\, \langle -, c_\eta, -\rangle \; c_{\eta\iota 0}\bar{e}_{\eta\iota 0}) \\
&\overset{\text{def}}{=} [\![\bar{\bar{e}}_{\eta\iota\times\eta\iota 1} = e_\eta \mapsto e_{\eta\iota 1} \mapsto \textstyle\sum_{e_{\eta\iota}, e_{\eta\iota 0}} \bar{e}_{\eta\iota 0}[e_{\eta\iota 0}] \\
&\quad \cdot \frac{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta e_\eta \overset{*}{\to} c_{\eta\iota}e_{\eta\iota} \; \ldots) \cdot \mathsf{P}_{\theta_{\text{G-L},d}}(c_{\eta\iota}e_{\eta\iota} \to c_{\eta\iota 0}e_{\eta\iota 0} \;\; c_{\eta\iota 1}e_{\eta\iota 1})}{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta e_\eta \overset{+}{\to} c_{\eta\iota 0}e_{\eta\iota 0} \; \ldots)}]\!]
\end{aligned} \tag{A.7}$$

$$\begin{aligned}
&= [\![\bar{\bar{e}}_{\eta\iota\times\eta\iota 1} = e_\eta \mapsto e_{\eta\iota 1} \mapsto \textstyle\sum_{e_{\eta\iota}} \mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta e_\eta \overset{*}{\to} c_{\eta\iota}e_{\eta\iota} \; \ldots) \\
&\quad \cdot \textstyle\sum_{l_{\eta\iota 0}, l_{\eta\iota 1}} \mathsf{P}_{\theta_{\text{M-L},d}}(c_{\eta\iota}e_{\eta\iota} \to l_{\eta\iota 0}c_{\eta\iota 0} \;\; l_{\eta\iota 1}c_{\eta\iota 1}) \\
&\quad \cdot \textstyle\sum_{e_{\eta\iota 0}} \mathsf{P}_{\theta_{\text{L}}}(e_{\eta\iota 0} \,|\, l_{\eta\iota 0} \; e_{\eta\iota}) \cdot \mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta e_\eta \overset{+}{\to} c_{\eta\iota 0}e_{\eta\iota 0} \; \ldots)^{-1} \\
&\quad \cdot \bar{e}_{\eta\iota 0}[e_{\eta\iota 0}] \cdot \mathsf{P}_{\theta_{\text{L}}}(e_{\eta\iota 1} \,|\, l_{\eta\iota 1} \; e_{\eta\iota})]\!]
\end{aligned} \tag{A.8}$$

$$\begin{aligned}
&= [\![\bar{\bar{e}}_{\eta\iota\times\eta\iota 1} = \bar{\bar{l}}^{*}_{\eta\times\eta\iota}(c_\eta, c_{\eta\iota}) \cdot \textstyle\sum_{l_{\eta\iota 0}, l_{\eta\iota 1}} \bar{\bar{m}}_{\text{L},d}(c_{\eta\iota} \to l_{\eta\iota 0}c_{\eta\iota 0} \; l_{\eta\iota 1}c_{\eta\iota 1}) \\
&\quad \cdot \mathrm{d}(\bar{\bar{l}}_{\eta\iota\times\eta\iota 0}(l_{\eta\iota 0}) \cdot \bar{\bar{l}}^{+}_{\eta\times\eta\iota 0}(c_\eta, c_{\eta\iota 0})^{-1} \cdot \bar{e}_{\eta\iota 0}) \cdot \bar{\bar{l}}_{\eta\iota\times\eta\iota 1}(l_{\eta\iota 1})]\!]
\end{aligned} \tag{A.9}$$

- Cross-element reduction (CER):

$$\begin{aligned}
&\mathsf{P}_{\theta_{\text{G-CER},d}}(c_{\eta\iota}, \bar{e}_{\eta\iota}, \mathbf{1} \,|\, \langle -, c_\eta, -\rangle \; \langle c'_{\eta\iota}, c_{\eta\iota\kappa}, \bar{\bar{e}}_{\eta\iota\times\eta\iota\kappa}\rangle) \\
&\overset{\text{def}}{=} [\![\bar{e}_{\eta\iota} = e_\eta \mapsto \sum_{e_{\eta\iota}, e_{\eta\iota\kappa}} \frac{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta e_\eta \overset{0}{\to} c_{\eta\iota}e_{\eta\iota})}{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta e_\eta \overset{*}{\to} c_{\eta\iota}e_{\eta\iota})} \cdot \bar{\bar{e}}_{\eta\iota\times\eta\iota\kappa}[e_{\eta\iota}, e_{\eta\iota\kappa}]]\!]
\end{aligned} \tag{A.10}$$

$$= [\![\bar{e}_{\eta\iota} = \bar{\bar{l}}^{0}_{\eta\times\eta\iota}(c_\eta, c_{\eta\iota}) \cdot \bar{\bar{l}}^{*}_{\eta\times\eta\iota}(c_\eta, c_{\eta\iota})^{-1} \cdot \bar{\bar{e}}'_{\eta\iota\times\eta\iota\kappa} \cdot \bar{\mathbf{1}}]\!] \tag{A.11}$$

- In-element reduction (IER):

$$\mathsf{P}_{\theta_{\text{G-CER},d}}(c_{\eta\iota}, \bar{e}_{\eta\iota}, \mathbf{0} \,|\, \langle -, c_\eta, - \rangle \; \langle c'_{\eta\iota}, c_{\eta\iota\kappa}, \bar{\bar{e}}_{\eta\iota\times\eta\iota\kappa} \rangle)$$

$$\stackrel{\text{def}}{=} [\![\bar{e}_{\eta\iota} = e_\eta \mapsto \sum_{e_{\eta\iota}, e'_{\eta\iota}, e_{\eta\iota\kappa}} \frac{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta e_\eta \xrightarrow{+} c_{\eta\iota} e_{\eta\iota})}{\mathsf{E}_{\theta_{\text{G-RL}^*,d}}(c_\eta e_\eta \xrightarrow{*} c_{\eta\iota} e_{\eta\iota})} \cdot \bar{\bar{e}}_{\eta\iota\times\eta\iota\kappa}[e_{\eta\iota}, e_{\eta\iota\kappa}]]\!] \tag{A.12}$$

$$= [\![\bar{e}_{\eta\iota} = \bar{\bar{l}}^{+}_{\eta\times\eta\iota}(c_\eta, c_{\eta\iota}) \cdot \bar{\bar{l}}^{*}_{\eta\times\eta\iota}(c_\eta, c_{\eta\iota})^{-1} \cdot \bar{\bar{e}}'_{\eta\iota\times\eta\iota\kappa} \cdot \bar{\mathbf{1}}]\!] \tag{A.13}$$

References

Gregory Aist, James Allen, Ellen Campana, Carlos Gallo, Scott Stoness, Mary Swift, and Michael Tanenhaus. Incremental understanding in human-computer dialogue and experimental evidence for advantages over nonincremental methods. In *Proc. DECALOG*, pages 149–154, 2007.

Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

Johan Bos. Predicate logic unplugged. In *Proceedings of the 10th Amsterdam Colloquium*, pages 133–143, 1996.

Eugene Charniak. Statistical parsing with a context-free grammar and word statistics. In *Fourteenth National Conference on Artificial Intelligence*, Providence, Rhode Island, 1997.

Noam Chomsky and George A. Miller. Introduction to the formal analysis of natural languages. In *Handbook of Mathematical Psychology*, pages 269–321. Wiley, 1963.

Michael Collins. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL '97)*, 1997.

Nelson Cowan. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24:87–185, 2001.

Simon Dennis, Dennis Mehay, and Srikar Yekollu. Predicting when words may appear: A connectionist model of sentence processing. In *The Proceedings of the Thirty First Conference of the Cognitive Science Society*. 2009.

David DeVault and Matthew Stone. Domain inference in incremental interpretation. In *Proc. ICoS*, pages 73–87, 2003.

Jeffrey L. Elman. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7:195–225, 1991.

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

Peter Gorniak and Deb Roy. Grounded semantic composition for visual scenes. *Journal of Artificial Intelligence Research*, 21:429–470, 2004.

Nicholas Haddock. Computational models of incremental semantic interpretation. *Language and Cognitive Processes*, 4:337–368, 1989.

Mark Johnson. Finite state approximation of constraint-based grammars using left-corner grammar transforms. In *Proceedings of COLING/ACL*, pages 619–623, Montreal, Canada, 1998.

Charles Martin and Christopher Riesbeck. Uniform parsing and inferencing for learning. In *Proceedings of AAAI*, pages 257–261, 1986.

Marshall R. Mayberry, III and Risto Miikkulainen. Incremental nonmonotonic parsing through semantic self-organization. In *Proceedings of the* 25^{th} *Annual Conference of the Cognitive Science Society*, pages 798–803, Boston, MA, 2003.

Chris Mellish. *Computer interpretation of natural language descriptions*. Wiley, New York, 1985.

George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63:81–97, 1956.

Kevin P. Murphy and Mark A. Paskin. Linear time inference in hierarchical HMMs. In *Proc. NIPS*, pages 833–840, Vancouver, BC, Canada, 2001.

William Schuler. Parsing with a bounded stack using a model-based right-corner transform. In *Proceedings of NAACL*, pages 344–352, Boulder, Colorado, 2009.

William Schuler, Samir AbdelRahman, Tim Miller, and Lane Schwartz. Toward a psycholinguistically-motivated model of language. In *Proceedings of COLING*, pages 785–792, Manchester, UK, August 2008.

William Schuler, Samir AbdelRahman, Tim Miller, and Lane Schwartz. Broad-coverage incremental parsing using human-like memory constraints. *Computational Linguistics*, 36(1), 2010.

Michael K. Tanenhaus, Michael J. Spivey-Knowlton, Kathy M. Eberhard, and Julie E. Sedivy. Integration of visual and linguistic information in spoken language comprehension. *Science*, 268:1632–1634, 1995.