An Analysis of Memory-based Processing Costs using Incremental Deep Syntactic Dependency Parsing*

Marten van Schijndel
The Ohio State University
vanschm@ling.osu.edu

Luan Nguyen
University of Minnesota
lnguyen@cs.umn.edu

William Schuler
The Ohio State University
schuler@ling.osu.edu

Abstract

Reading experiments using naturalistic stimuli have shown unanticipated facilitations for completing center embeddings when frequency effects are factored out. To eliminate possible confounds due to surface structure, this paper introduces a processing model based on deep syntactic dependencies. Results on eye-tracking data indicate that completing deep syntactic embeddings yields significantly more facilitation than completing surface embeddings.

1 Introduction

Self-paced reading and eye-tracking experiments have often been used to support theories about inhibitory effects of working memory operations in sentence processing (Just and Carpenter, 1992; Gibson, 2000; Lewis and Vasishth, 2005), but it is possible that many of these effects can be explained by frequency (Jurafsky, 1996; Hale, 2001; Karlsson, 2007). Experiments on large naturalistic text corpora (Demberg and Keller, 2008; Wu et al., 2010; van Schijndel and Schuler, 2013) have shown significant memory effects at the ends of center embeddings when frequency measures have been included as separate factors, but these memory effects have been facilitatory rather than inhibitory.

Some of the memory-based measures that produce these facilitatory effects (Wu et al., 2010; van Schijndel and Schuler, 2013) are defined in terms of initiation and integration of *connected components* of syntactic structure,[1] with the presumption that referents that belong to the same connected component may cue one another using content-based features, while those that do not must rely on noisier temporal features that just encode how recently a referent was accessed. These measures, based on left-corner parsing processes (Johnson-Laird, 1983; Abney and Johnson, 1991), abstract counts of unsatisfied dependencies from noun or verb referents (Gibson, 2000) to cover all syntactic dependencies, motivated by observations of Demberg and Keller (2008) and Kwon et al. (2010) of the inadequacies of Gibson's narrower measure.

But these experiments use naturalistic stimuli without constrained manipulations and therefore might be susceptible to confounds. It is possible that the purely phrase-structure-based connected components used previously may ignore some integration costs associated with filler-gap constructions, making them an unsuitable generalization of Gibson-style dependencies. It is also possible that the facilitatory effect for integration operations in naturally-occurring stimuli may be driven by syntactic center embeddings that arise from modifiers (e.g. *The CEO sold [[the shares] of the company]*), which do not require any dependencies to be deferred, but which might be systematically under-predicted by frequency measures, producing a confound with memory measures when frequency measures are residualized out.

In order to eliminate possible confounds due to exclusion of unbounded dependencies in filler-gap constructions, this paper evaluates a processing model that calculates connected components on deep syntactic dependency structures rather than surface phrase structure trees. This model accounts unattached fillers and gaps as belonging to separate connected components, and therefore performs additional initiation and integration op-

*Thanks to Micha Elsner and three anonymous reviewers for their feedback. This work was funded by an Ohio State University Department of Linguistics Targeted Investment for Excellence (TIE) grant for collaborative interdisciplinary projects conducted during the academic year 2012–13.

[1] Graph theoretically, the set of connected components of a graph $\langle V, E \rangle$ is the set of maximal subsets of it $\{\langle V_1, E_1 \rangle, \langle V_2, E_2 \rangle, ...\}$ such that any pair of vertices in each V_i can be connected by edges in the corresponding E_i.

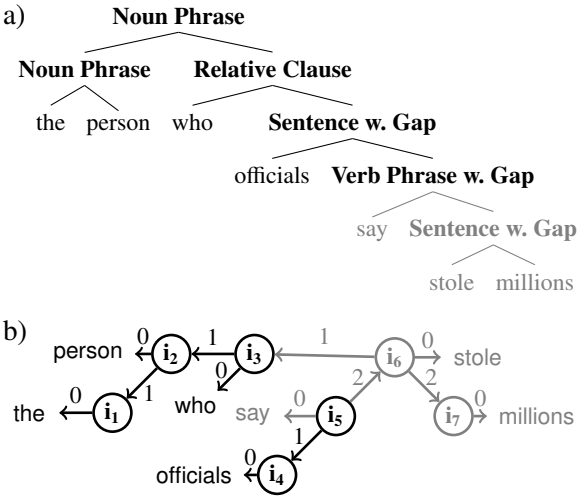

Figure 1: Graphical representation of (a) a single connected component of surface syntactic phrase structure corresponding to (b) two connected components of deep syntactic dependency structure for the noun phrase *the person who officials say stole millions*, prior to the word *say*. Connections established prior to the word *say* are shown in black; subsequent connections are shown in gray.

erations in filler-gap constructions as hypothesized by Gibson (2000) and others. Then, in order to control for possible confounds due to modifier-induced center embedding, this refined model is applied to two partitions of an eye-tracking corpus (Kennedy et al., 2003): one consisting of sentences containing only non-modifier center embeddings, in which dependencies are deferred, and the other consisting of sentences containing no center embeddings or containing center embeddings arising from attachment of final modifiers, in which no dependencies are deferred. Processing this partitioned corpus with deep syntactic connected components reveals a significant increase in facilitation in the non-modifier partition, which lends credibility to the observation of negative integration cost in processing naturally-occurring sentences.

2 Connected Components

The experiments described in this paper evaluate whether inhibition and facilitation in reading correlate with operations in a hierarchic sequential prediction model that initiate and integrate connected components of hypothesized syntactic structure during incremental parsing. The model used in these experiments refines previous connected component models by allowing fillers and gaps to occur in separate connected components of a deep syntactic dependency graph (Mel'čuk, 1988; Kintsch, 1988), even when they belong to the same connected component when defined on surface structure.

For example, the surface syntactic phrase structure and deep syntactic dependency structure for the noun phrase *the person who officials say stole millions* are shown in Figure 1.[2] Notice that after the word *officials*, there is only one connected component of surface syntactic phrase structure (from the root noun phrase to the verb phrase with gap), but two disjoint connected components of deep syntactic dependency structure (one ending at $\mathbf{i_3}$, and another at $\mathbf{i_5}$). Only the deep syntactic dependency structure corresponds to familiar (Just and Carpenter, 1992; Gibson, 1998) notions of how memory is used to store deferred dependencies in filler-gap constructions. The next section will describe a generalized categorial grammar, which (i) can be viewed as context-free, to seed a latent-variable probabilistic context-free grammar to accurately derive parses of filler-gap constructions, and (ii) can be viewed as a deep syntactic dependency grammar, defining dependencies for connected components in terms of function applications.

3 Generalized Categorial Grammar

In order to evaluate memory effects for hypothesizing unbounded dependencies between referents of fillers and referents of clauses containing gaps, a memory-based processor must define connected components in terms of deep syntactic dependencies (including unbounded dependencies) rather than in terms of surface syntactic phrase structure trees. To do this, at least some phrase structure edges must be removed from the set of connections that define a connected component.

Because these unbounded dependencies are not represented locally in the original Treebank format, probabilities for operations on these modified

[2]Following Mel'čuk (1988) and Kintsch (1988), the graphical dependency structure adopted here uses positionally-defined labels ('0' for the predicate label, '1' for the first argument ahead of a predicate, '2' for the last argument behind, etc.) but includes unbounded dependencies between referents of fillers and referents of clauses containing gaps. It is assumed that semantically-labeled structures would be isomorphic to the structures defined here, but would generalize across alternations such as active and passive constructions, for example.

connected components are trained on a corpus annotated with generalized categorial grammar dependencies for 'gap' arguments at all categories that subsume a gap (Nguyen et al., 2012). This representation is similar to the HPSG-like representation used by Hale (2001) and Lewis and Vasishth (2005), but has a naturally-defined dependency structure on which to calculate connected components. This generalized categorial grammar is then used to identify the first sign that introduces a gap, at which point a deep syntactic connected component containing the filler can be encoded (stored), and a separate deep syntactic connected component for a clause containing a gap can be initiated.

A generalized categorial grammar (Bach, 1981) consists of a set U of primitive category types; a set O of type-constructing operators allowing a recursive definition of a set of categories $C =_{\text{def}} U \cup (C \times O \times C)$; a set X of vocabulary items; a mapping M from vocabulary items in X to semantic functions with category types in C; and a set R of inference rules for deriving functions with category types in C from other functions with category types in C. Nguyen et al. (2012) use primitive category types for clause types (e.g. **V** for finite verb-headed clause, **N** for noun phrase or nominal clause, **D** for determiners and possessive clauses, etc.), and use the generalized set of type-constructing operators to characterize not only function application dependencies between arguments immediately ahead of and behind a functor (**-a** and **-b**, corresponding to '\' and '/' in Ajdukiewicz-Bar-Hillel categorial grammars), but also long-distance dependencies between fillers and categories subsuming gaps (**-g**), dependencies between relative pronouns and antecedent modificands of relative clauses (**-r**), and dependencies between interrogative pronouns and their arguments (**-i**), which remain unsatisfied in derivations but function to distinguish categories for content and polar questions. A lexicon can then be defined in M to introduce lexical dependencies and obligatory pronominal dependencies using numbered functions for predicates and deep syntactic arguments, for example:

$$\begin{aligned}
\text{the} &\Rightarrow (\lambda_i\,(0\,i){=}\text{the}) : \mathbf{D}\\
\text{person} &\Rightarrow (\lambda_i\,(0\,i){=}\text{person}) : \mathbf{N\text{-}aD}\\
\text{who} &\Rightarrow (\lambda_{k\,i}\,(0\,i){=}\text{who} \wedge (1\,i){=}k) : \mathbf{N\text{-}rN}\\
\text{officials} &\Rightarrow (\lambda_i\,(0\,i){=}\text{officials}) : \mathbf{N}
\end{aligned}$$

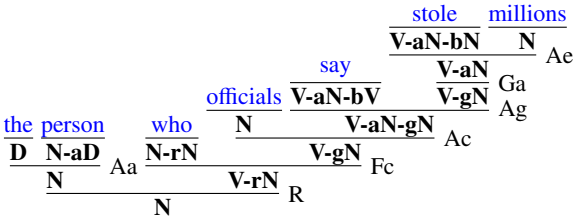

Figure 2: Example categorization of the noun phrase *the person who officials say stole millions.*

$$\begin{aligned}
\text{say} &\Rightarrow (\lambda_i\,(0\,i){=}\text{say}) : \mathbf{V\text{-}aN\text{-}bV}\\
\text{stole} &\Rightarrow (\lambda_i\,(0\,i){=}\text{stole}) : \mathbf{V\text{-}aN\text{-}bN}\\
\text{millions} &\Rightarrow (\lambda_i\,(0\,i){=}\text{millions}) : \mathbf{N}
\end{aligned}$$

Inference rules in R are then defined to compose arguments and modifiers and propagate gaps. Arguments g of type d ahead of functors h of type c-**a**d are composed by passing non-local dependencies $\psi \in \{\mathbf{-g}, \mathbf{-i}, \mathbf{-r}\} \times C$ from premises to conclusion in all combinations:

$$g{:}d \;\; h{:}\,c\text{-}\mathbf{a}d \Rightarrow (f_{c\text{-}\mathbf{a}d}\,g\,h){:}\,c \qquad \text{(Aa)}$$
$$g{:}d\psi \;\; h{:}\,c\text{-}\mathbf{a}d \Rightarrow \lambda_k\,(f_{c\text{-}\mathbf{a}d}\,(g\,k)\,h){:}\,c\psi \qquad \text{(Ab)}$$
$$g{:}d \;\; h{:}\,c\text{-}\mathbf{a}d\psi \Rightarrow \lambda_k\,(f_{c\text{-}\mathbf{a}d}\,g\,(h\,k)){:}\,c\psi \qquad \text{(Ac)}$$
$$g{:}d\psi \;\; h{:}\,c\text{-}\mathbf{a}d\psi \Rightarrow \lambda_k\,(f_{c\text{-}\mathbf{a}d}\,(g\,k)\,(h\,k)){:}\,c\psi \qquad \text{(Ad)}$$

Similar rules compose arguments behind functors:

$$g{:}\,c\text{-}\mathbf{b}d \;\; h{:}d \Rightarrow (f_{c\text{-}\mathbf{b}d}\,g\,h){:}\,c \qquad \text{(Ae)}$$
$$g{:}\,c\text{-}\mathbf{b}d\psi \;\; h{:}d \Rightarrow \lambda_k\,(f_{c\text{-}\mathbf{b}d}\,(g\,k)\,h){:}\,c\psi \qquad \text{(Af)}$$
$$g{:}\,c\text{-}\mathbf{b}d \;\; h{:}d\psi \Rightarrow \lambda_k\,(f_{c\text{-}\mathbf{b}d}\,g\,(h\,k)){:}\,c\psi \qquad \text{(Ag)}$$
$$g{:}\,c\text{-}\mathbf{b}d\psi \;\; h{:}d\psi \Rightarrow \lambda_k\,(f_{c\text{-}\mathbf{b}d}\,(g\,k)\,(h\,k)){:}\,c\psi \qquad \text{(Ah)}$$

These rules use composition functions $f_{c\text{-}\mathbf{a}d}$ and $f_{c\text{-}\mathbf{b}d}$ for initial and final arguments, which define dependency edges numbered v from referents of predicate functors i to referents of arguments j, where v is the number of unsatisfied arguments $\varphi_1 ... \varphi_v \in \{\mathbf{-a}, \mathbf{-b}\} \times C$ in a category label:

$$f_{u\varphi_{1..v-1}\text{-}\mathbf{a}c} \stackrel{\text{def}}{=} \lambda_{g\,h\,i}\,\exists_j\,(v\,i){=}j \wedge (g\,j) \wedge (h\,i) \qquad \text{(1a)}$$
$$f_{u\varphi_{1..v-1}\text{-}\mathbf{b}c} \stackrel{\text{def}}{=} \lambda_{g\,h\,i}\,\exists_j\,(v\,i){=}j \wedge (g\,i) \wedge (h\,j) \qquad \text{(1b)}$$

R also contains inference rules to compose modifier functors g of type u-**a**d ahead of modificands h of type d:

$$g{:}\,u\text{-}\mathbf{a}d \;\; h{:}c \Rightarrow (f_{\text{IM}}\,g\,h){:}c \qquad \text{(Ma)}$$
$$g{:}\,u\text{-}\mathbf{a}d\psi \;\; h{:}c \Rightarrow \lambda_k\,(f_{\text{IM}}\,(g\,k)\,h){:}c\psi \qquad \text{(Mb)}$$
$$g{:}\,u\text{-}\mathbf{a}d \;\; h{:}c\psi \Rightarrow \lambda_k\,(f_{\text{IM}}\,g\,(h\,k)){:}c\psi \qquad \text{(Mc)}$$

$$\frac{\exists_{i^1\,j^1\ldots\,i^\ell\,j^\ell}\ \ldots\ \wedge\ (g^\ell{:}c/d\,\{j^\ell\}\,i^\ell)\quad x_t}{\exists_{i^1\,j^1\ldots\,i^\ell}\ \ldots\ \wedge\ ((g^\ell f){:}c\,i^\ell)}\ x_t \Rightarrow f{:}d \qquad \text{(–Fa)}$$

$$\frac{\exists_{i^1\,j^1\ldots\,i^\ell\,j^\ell}\ \ldots\ \wedge\ (g^\ell{:}c/d\,\{j^\ell\}\,i^\ell)\quad x_t}{\exists_{i^1\,j^1\ldots\,i^\ell\,j^\ell\,i^{\ell+1}}\ \ldots\ \wedge\ (g^\ell{:}c/d\,\{j^\ell\}\,i^\ell)\ \wedge\ (f{:}e\,i^{\ell+1})}\ x_t \Rightarrow f{:}e \qquad \text{(+Fa)}$$

$$\frac{\exists_{i^1\,j^1\ldots\,i^{\ell-1}\,j^{\ell-1}\,i^\ell}\ \ldots\ \wedge\ (g^\ell{:}d\,i^\ell)}{\exists_{i^1\,j^1\ldots\,i^\ell\,j^\ell}\ \ldots\ \wedge\ ((f\,g^\ell){:}c/e\,\{j^\ell\}\,i^\ell)}\ \begin{cases} g{:}d\ h{:}e \Rightarrow (f\ g\ h){:}c & \text{or}\\ g{:}d\ h{:}e \Rightarrow \lambda_k(f\ (g\ k)\ h){:}c & \text{or}\\ g{:}d\ h{:}e \Rightarrow \lambda_k(f\ g\ (h\ k)){:}c & \text{or}\\ g{:}d\ h{:}e \Rightarrow \lambda_k(f\ (g\ k)\ (h\ k)){:}c \end{cases} \qquad \text{(–La)}$$

$$\frac{\exists_{i^1\,j^1\ldots\,i^{\ell-1}\,j^{\ell-1}\,i^\ell}\ \ldots\ \wedge\ (g^{\ell-1}{:}a/c\,\{j^{\ell-1}\}\,i^{\ell-1})\ \wedge\ (g^\ell{:}d\,i^\ell)}{\exists_{i^1\,j^1\ldots\,i^{\ell-1}\,j^{\ell-1}}\ \ldots\ \wedge\ (g^{\ell-1}\circ(f\,g^\ell){:}a/e\,\{j^{\ell-1}\}\,i^{\ell-1})}\ \begin{cases} g{:}d\ h{:}e \Rightarrow (f\ g\ h){:}c & \text{or}\\ g{:}d\ h{:}e \Rightarrow \lambda_k(f\ (g\ k)\ h){:}c & \text{or}\\ g{:}d\ h{:}e \Rightarrow \lambda_k(f\ g\ (h\ k)){:}c & \text{or}\\ g{:}d\ h{:}e \Rightarrow \lambda_k(f\ (g\ k)\ (h\ k)){:}c \end{cases} \qquad \text{(+La)}$$

Figure 3: Basic processing productions of a right-corner parser.

$$g{:}\,u\text{-}\mathbf{a}d\psi\ \ h{:}c\psi \Rightarrow \lambda_k\,(f_{\text{IM}}\,(g\,k)\,(h\,k)){:}c\psi \qquad \text{(Md)}$$

or for modifier functors behind a modificand:

$$g{:}c\ \ h{:}\,u\text{-}\mathbf{a}d \Rightarrow (f_{\text{FM}}\,g\,h){:}c \qquad \text{(Me)}$$
$$g{:}c\psi\ \ h{:}\,u\text{-}\mathbf{a}d \Rightarrow \lambda_k\,(f_{\text{FM}}\,(g\,k)\,h){:}c\psi \qquad \text{(Mf)}$$
$$g{:}c\ \ h{:}\,u\text{-}\mathbf{a}d\psi \Rightarrow \lambda_k\,(f_{\text{FM}}\,g\,(h\,k)){:}c\psi \qquad \text{(Mg)}$$
$$g{:}c\psi\ \ h{:}\,u\text{-}\mathbf{a}d\psi \Rightarrow \lambda_k\,(f_{\text{FM}}\,(g\,k)\,(h\,k)){:}c\psi \qquad \text{(Mh)}$$

These rules use composition functions f_{IM} and f_{FM} for initial and final modifiers, which define dependency edges numbered '1' from referents of modifier functors i to referents of modificands j:

$$f_{\text{IM}} \stackrel{\text{def}}{=} \lambda_{g\,h\,j}\,\exists_i\,(1\,i){=}j \wedge (g\,i) \wedge (h\,j) \qquad \text{(2a)}$$
$$f_{\text{FM}} \stackrel{\text{def}}{=} \lambda_{g\,h\,j}\,\exists_i\,(1\,i){=}j \wedge (g\,j) \wedge (h\,i) \qquad \text{(2b)}$$

R also contains inference rules for hypothesizing gaps **-g**d for arguments and modifiers:[3]

$$g{:}\,c\text{-}\mathbf{a}d \Rightarrow \lambda_k\,(f_{c\text{-}\mathbf{a}d}\,\{k\}\,g){:}\,c\text{-}\mathbf{g}d \qquad \text{(Ga)}$$
$$g{:}\,c\text{-}\mathbf{b}d \Rightarrow \lambda_k\,(f_{c\text{-}\mathbf{a}d}\,\{k\}\,g){:}\,c\text{-}\mathbf{g}d \qquad \text{(Gb)}$$
$$g{:}c \Rightarrow \lambda_k\,(f_{\text{IM}}\,\{k\}\,g){:}c\text{-}\mathbf{g}d \qquad \text{(Gc)}$$

and for attaching fillers e, d-**r**e, d-**i**e as gaps **-g**d:

$$g{:}e\ \ h{:}\,c\text{-}\mathbf{g}d \Rightarrow \lambda_i\,\exists_j\,(g\,i) \wedge (h\,i\,j){:}e \qquad \text{(Fa)}$$
$$g{:}d\text{-}\mathbf{r}e\ \ h{:}\,c\text{-}\mathbf{g}d \Rightarrow \lambda_{k\,j}\,\exists_i\,(g\,k\,i) \wedge (h\,i\,j){:}\,c\text{-}\mathbf{r}e \qquad \text{(Fb)}$$
$$g{:}d\text{-}\mathbf{i}e\ \ h{:}\,c\text{-}\mathbf{g}d \Rightarrow \lambda_{k\,j}\,\exists_i\,(g\,k\,i) \wedge (h\,i\,j){:}\,c\text{-}\mathbf{i}e \qquad \text{(Fc)}$$

[3] Since these unary inferences perform no explicit composition, they are defined to use only initial versions composition functions $f_{c\text{-}\mathbf{a}d}$ and f_{IM}.

and for attaching modificands as antecedents of relative pronouns:

$$g{:}e\ \ h{:}c\text{-}\mathbf{r}d \Rightarrow \lambda_i\,\exists_j\,(g\,i) \wedge (h\,i\,j){:}e \qquad \text{(R)}$$

An example derivation of the noun phrase *the person who officials say stole millions* using these rules is shown in Figure 2. The semantic expression produced by this derivation consists of a conjunction of terms defining the edges in the graph shown in Figure 1b.

This GCG formulation captures many of the insights of the HPSG-like context-free filler-gap notation used by Hale (2001) or Lewis and Vasishth (2005): inference rules with adjacent premises can be cast as context-free grammars and weighted using probabilities, which allow experiments to calculate frequency measures for syntactic constructions. Applying a latent variable PCFG trainer (Petrov et al., 2006) to this formulation was shown to yield state-of-the-art accuracy for recovery of unbounded dependencies (Nguyen et al., 2012). Moreover, the functor-argument dependencies in a GCG define deep syntactic dependency graphs for all derivations, which can be used in incremental parsing to calculate connected components for memory-based measures.

4 Incremental Processing

In order to obtain measures of memory operations used in incremental processing, these GCG inference rules are combined into a set of parser

$$\frac{\exists_{i^1\,j^1..\,i^n\,j^n..\,i^\ell\,j^\ell}\ ...\ \wedge\ (g^n{:}y/z\psi\,\{j^n\}\,i^n)\ \wedge\ ...\ \wedge\ (g^\ell{:}c/d\,\{j^\ell\}\,i^\ell)\quad x_t}{\exists_{i^1\,j^1..\,i^n\,j^n..\,i^\ell}\ ...\ \wedge\ (g^n{:}y/z\psi\,\{j^n\}\,i^n)\ \wedge\ ...\ \wedge\ ((g^\ell(f'\{j^n\}f)){:}c\,i^\ell)}\ x_t \Rightarrow \lambda_k(f'\{k\}\,f){:}d$$

(–Fb)

$$\frac{\exists_{i^1\,j^1..\,i^n\,j^n..\,i^\ell\,j^\ell}\ ...\ \wedge\ (g^n{:}y/z\psi\,\{j^n\}\,i^n)\ \wedge\ ...\ \wedge\ (g^\ell{:}c/d\,\{j^\ell\}\,i^\ell)\quad x_t}{\exists_{i^1\,j^1..\,i^n\,j^n..\,i^\ell\,j^\ell\,i^{\ell+1}}\ ...\ \wedge\ (g^n{:}y/z\psi\,\{j^n\}\,i^n)\ \wedge\ ...\ \wedge\ (g^\ell{:}c/d\,\{j^\ell\}\,i^\ell)\ \wedge\ ((f'\{j^n\}f){:}e\,i^{\ell+1})}\ x_t \Rightarrow \lambda_k(f'\{k\}\,f){:}e$$

(+Fb)

$$\frac{\exists_{i^1\,j^1..\,i^n\,j^n..\,i^{\ell-1}\,j^{\ell-1}\,i^\ell}\ ...\ \wedge\ (g^n{:}y/z\psi\,\{j^n\}\,i^n)\ \wedge\ ...\ \wedge\ (g^\ell{:}d\,i^\ell)}{\exists_{i^1\,j^1..\,i^n\,j^n..\,i^\ell\,j^\ell}\ ...\ \wedge\ (g^n{:}y/z\psi\,\{j^n\}\,i^n)\ \wedge\ ...\ \wedge\ ((fg^\ell)\circ(f'\{j^n\}){:}c\psi/e\,\{j^\ell\}\,i^\ell)}$$

$$g{:}d\ h{:}e \Rightarrow \lambda_k(fg\,(f'\{k\}\,h)){:}c\psi \quad \text{(–Lb)}$$

$$\frac{\exists_{i^1\,j^1..\,i^n\,j^n..\,i^{\ell-1}\,j^{\ell-1}\,i^\ell}\ ...\ \wedge\ (g^n{:}y/z\psi\,\{j^n\}\,i^n)\ \wedge\ ...\ \wedge\ (g^{\ell-1}{:}a/c\psi\,\{j^{\ell-1}\}\,i^{\ell-1})\ \wedge\ (g^\ell{:}d\,i^\ell)}{\exists_{i^1\,j^1..\,i^n\,j^n..\,i^{\ell-1}\,j^{\ell-1}}\ ...\ \wedge\ (g^n{:}y/z\psi\,\{j^n\}\,i^n)\ \wedge\ ...\ \wedge\ (g^{\ell-1}\circ(fg^\ell)\circ(f'\{j^n\}){:}a/e\,\{j^{\ell-1}\}\,i^{\ell-1})}$$

$$g{:}d\ h{:}e \Rightarrow \lambda_k(fg\,(f'\{k\}\,h)){:}c\psi \quad \text{(+Lb)}$$

Figure 4: Additional processing productions for attaching a referent of a filler j^n as the referent of a gap.

productions, similar to those of the 'right corner' parser of van Schijndel and Schuler (2013), except that instead of recognizing shallow hierarchical sequences of connected components of surface structure, the parser recognizes shallow hierarchical sequences of connected components of deep syntactic dependencies. This parser exploits the observation (van Schijndel et al., in press) that left-corner parsers and their variants do not need to initiate or integrate more than one connected component at each word. These two operations are then augmented with rules to introduce fillers and attach fillers as gaps.

This parser is defined on *incomplete connected component states* which consist of an *active sign* (with a semantic referent and syntactic form or category) lacking an *awaited sign* (also with a referent and category) yet to come. Semantic functions of active and awaited signs are simplified to denote only sets of referents, with gap arguments (λ_k) stripped off and handled by separate connected components. Incomplete connected components, therefore, always denote semantic functions from sets of referents to sets of referents.

This paper will notate semantic functions of connected components using variables g and h, incomplete connected component categories as c/d (consisting of an active sign of category c and an awaited sign of category d), and associations between them as $g{:}c/d$. The semantic representation used here is simply a deep syntactic dependency structure, so a connected component function is satisfied if it holds for some output referent i given input referent j. This can be notated $\exists_{i\,j}\,(g{:}c/d\,\{j\}\,i)$, where the set $\{j\}$ is equivalent to $(\lambda_{j'}\ j'{=}j)$. Connected component functions that have a common referent j can then be composed into larger connected components:[4]

$$\exists_{ijk}\,(g\,\{j\}\,i)\ \wedge\ (h\,\{k\}\,j)\ \ \Leftrightarrow\ \ \exists_{ij}\,(g{\circ}h\,\{k\}\,i) \quad (3)$$

Hierarchies of ℓ connected components can be represented as conjunctions: $\exists_{i^1\,j^1...\,i^\ell\,j^\ell}\,(g^1{:}c^1/d^1\,\{j^1\}\,i^1)\wedge ...\wedge(g^\ell{:}c^\ell/d^\ell\,\{j^\ell\}\,i^\ell)$. This allows constraints such as unbounded dependencies between referents of fillers and referents of clauses containing gaps to be specified across connected components by simply plugging variables for filler referents into argument positions for gaps.

A nondeterministic incremental parser can now be defined as a deductive system, given an input sequence consisting of an initial connected component state of category $\mathbf{T}/\mathbf{T}$, corresponding to an existing discourse context, followed by a sequence of observations $x_1, x_2, \ldots$, processed in time order. As each x_t is encountered, it is connected to an existing connected component or it introduces a new disjoint component using the productions shown in Figures 3, 4, and 5.

[4]These are connected components of dependency structure resulting from one or more composition functions being composed, with each function's output as the previous function's second argument. This uses a standard definition of function composition: $((f \circ g)\,x) = (f\,(g\,x))$.

$$\frac{\exists_{i^1\,j^1_{..}\,i^{\ell-1}\,j^{\ell-1}\,i^\ell}\ \ldots\ \wedge\ (g^\ell{:}d\,i^\ell)}{\exists_{i^1\,j^1_{..}\,i^\ell\,j^\ell}\ \ldots\ \wedge\ ((f\,g^\ell)\circ(\lambda_{h\,k\,i}\,(h\,k)){:}a/e\psi\,\{j^\ell\}\,i^\ell)}\ \ g{:}d\ h{:}e\psi \Rightarrow (f\,g\,h){:}c \qquad \text{(–Lc)}$$

$$\frac{\exists_{i^1\,j^1_{..}\,i^{\ell-1}\,j^{\ell-1}\,i^\ell}\ \ldots\ \wedge\ (g^{\ell-1}{:}a/c\,\{j^{\ell-1}\}\,i^{\ell-1})\ \wedge\ (g^\ell{:}d\,i^\ell)}{\exists_{i^1\,j^1_{..}\,i^{\ell-1}\,j^{\ell-1}}\ \ldots\ \wedge\ (g^{\ell-1}\circ(f\,g^\ell)\circ(\lambda_{h\,k\,i}\,(h\,k)){:}a/e\psi\,\{j^{\ell-1}\}\,i^{\ell-1})}\ \ g{:}d\ h{:}e\psi \Rightarrow (f\,g\,h){:}c \qquad \text{(+Lc)}$$

$$\frac{\exists_{i^1\,j^1_{..}\,i^\ell\,j^\ell}\ \ldots\ \wedge\ (g^{\ell-1}{:}c/d\psi\,\{j^{\ell-1}\}\,i^{\ell-1})\ \wedge\ (g^\ell{:}d\psi/e\,\{j^\ell\}\,i^\ell)}{\exists_{i^1\,j^1_{..}\,i^{\ell-1}\,j^{\ell-1}}\ \ldots\ \wedge\ (g^{\ell-1}\circ(\lambda_{h\,i}\exists_j(h\,j))\circ g^\ell{:}c/e\,\{j^{\ell-1}\}\,i^{\ell-1})} \qquad \text{(+N)}$$

Figure 5: Additional processing productions for hypothesizing filler-gap attachment.

Operations on dependencies that can be derived from surface structure (see Figure 3) are taken directly from van Schijndel and Schuler (2013). First, if an observation x_t can immediately fill the awaited sign of the last connected component $g^\ell{:}c/d$, it is hypothesized to do so, turning this incomplete connected component into a complete connected component $(g^\ell\,f){:}c$ (Production –Fa); or if the observation can serve as an initial sub-sign of this awaited sign, it is hypothesized to form a new complete sign $f{:}e$ in a new component with x_t as its first observation (Production +Fa). Then, if either of these resulting complete signs $g^\ell{:}d$ can immediately attach as an initial child of the awaited sign of the most recent connected component $g^{\ell-1}{:}a/c$, it is hypothesized to merge and extend this connected component, with x_t as the last observation of the completed connected component (Production +La); or if it can serve as an initial sub-sign of this awaited sign, it is hypothesized to remain disjoint and form its own connected component (Production –La). The side conditions of La productions are defined to unpack gap propagation (instances of λ_k that distinguish rules Aa–h and Ma–h) from the inference rules in Section 3, because this functionality will be replaced with direct substitution of referent variables into subordinate semantic functions, below.

The Nguyen et al. (2012) GCG was defined to pass up unbounded dependencies, but in incremental deep syntactic dependency processing, unbounded dependencies are accounted as separate connected components. When hypothesizing an unbounded dependency, the processing model simply cues the active sign of a previous connected component containing a filler without completing the current connected component. The four +F, –F, +L, and –L operations are therefore combined with applications of unary rules Ga–c for hypothesizing referents as fillers for gaps (providing f' in the equations in Figure 4). Productions –Fb and +Fb fill gaps in initial children, and Productions –Lb and +Lb fill gaps in final children. Note that the Fb and Lb productions apply to the same types of antecedents as Fa and La productions respectively, so members of these two sets of productions cannot be applied together.

Applications of rules Fa–c and R for introducing fillers are applied to store fillers as existentially quantified variable values in Lc productions (see Figure 5). These Lc productions apply to the same type of antecedent as La and Lb productions, so these also cannot be applied together.

Finally, connected components separated by gaps which are no longer hypothesized (ψ) are reattached by a +N production. This +N production may then be paired with a –N production which yields its antecedent unchanged as a consequent. These N productions apply to antecedents and consequents of the same type, so they may be applied together with one F and one L production, but since the +N production removes in its consequent a ψ argument required in its antecedent, it may not apply more than once in succession (and applying the –N production more than once in succession has no effect).

An incremental derivation of the noun phrase *the person who officials say stole millions,* using these productions, is shown in Figure 6.

5 Evaluation

The F, L, and N productions defined in the previous section can be made probabilistic by first computing a probabilistic context-free grammar (PCFG) from a tree-annotated corpus, then transforming that PCFG model into a model of probabilities over incremental parsing operations using a grammar transform (Schuler, 2009). This allows the intermediate PCFG to be optimized using an existing PCFG-based latent variable trainer

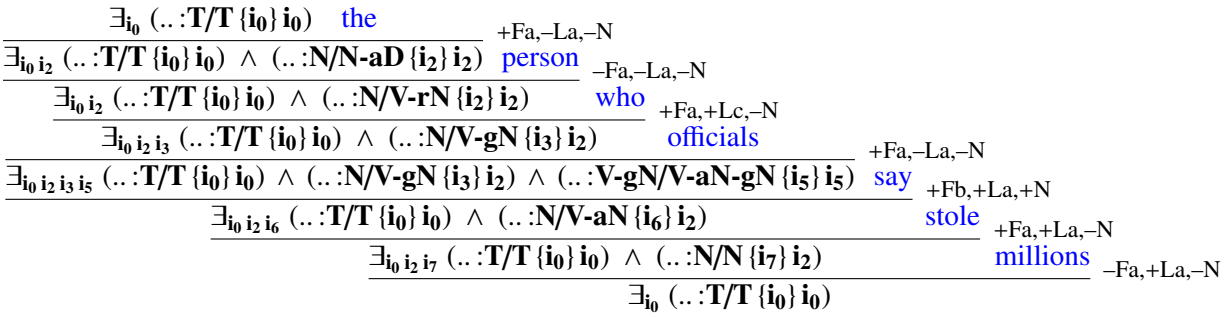

Figure 6: Derivation of *the person who officials say stole millions,* showing connected components with unique referent variables (calculated according to the equations in Section 4). Semantic functions are abbreviated to '..' for readability. This derivation yields the following lexical relations: (0 $\mathbf{i_1}$)=the, (0 $\mathbf{i_2}$)=person, (0 $\mathbf{i_3}$)=who, (0 $\mathbf{i_4}$)=officials, (0 $\mathbf{i_5}$)=say, (0 $\mathbf{i_6}$)=stole, (0 $\mathbf{i_7}$)=millions, and the following argument relations: (1 $\mathbf{i_2}$)=$\mathbf{i_1}$, (1 $\mathbf{i_3}$)=$\mathbf{i_2}$, (1 $\mathbf{i_5}$)=$\mathbf{i_4}$, (2 $\mathbf{i_5}$)=$\mathbf{i_6}$, (1 $\mathbf{i_6}$)=$\mathbf{i_3}$, (2 $\mathbf{i_6}$)=$\mathbf{i_7}$.

(Petrov et al., 2006). When applied to the output of this trainer, this transform has been shown to produce comparable accuracy to that of the original Petrov et al. (2006) CKY parser (van Schijndel et al., 2012). The transform used in these experiments diverges from that of Schuler (2009), in that the probability associated with introducing a gap in a filler-gap construction is reallocated from a –F–L operation to a +F–L operation (to encode the previously most subordinate connected component with the filler as its awaited sign and begin a new disjoint connected component), and the probability associated with resolving such a gap is reallocated from an implicit –N operation to a +N operation (to integrate the connected component containing the gap with that containing the filler).

In order to verify that the modifications to the transform correctly reallocate probability mass for gap operations, the goodness of fit to reading times of a model using this modified transform is compared against the publicly-available baseline model from van Schijndel and Schuler (2013), which uses the original Schuler (2009) transform.[5]

To ensure a valid comparison, both parsers are trained on a GCG-reannotated version of the Wall Street Journal portion of the Penn Treebank (Marcus et al., 1993) before being fit to reading times using linear mixed-effects models (Baayen et al., 2008).[6] This evaluation focuses on the processing that can be done up to a given point in a sentence. In human subjects, this processing includes both immediate lexical access and regressions that aid in the integration of new information, so the reading times of interest in this evaluation are log-transformed go-past durations.[7]

The first and last word of each line in the Dundee corpus, words not observed at least 5 times in the WSJ training corpus, and fixations after long saccades (>4 words) are omitted from the evaluation to filter out wrap-up effects, parser inaccuracies, and inattention and track loss of the eyetracker. The following predictors are centered and used in each baseline model: sentence position, word length, whether or not the previous or next word were fixated upon, and unigram and bigram probabilities.[8] Then each of the following predictors is residualized off each baseline before being centered and added to it to help residualize the next factor: length of the go-past region, cumulative total surprisal, total surprisal (Hale, 2001), and cumulative entropy reduction (Hale, 2003).[9] All 2-way interactions between these effects are

[5]The models used here also use random slopes to reduce their variance, which makes them less anticonservative.

[6]The models are built using *lmer* from the *lme4* R package (Bates et al., 2011; R Development Core Team, 2010).

[7]Go-past durations are calculated by summing all fixations in a region of text, including regressions, until a new region is fixated, which accounts for additional processing that may take place after initial lexical access, but before the next region is processed. For example, if one region ends at word 5 in a sentence, and the next fixation lands on word 8, then the go-past region consists of words 6-8 while go-past duration sums all fixations until a fixation occurs after word 8. Log-transforming eye movements and fixations may make their distributions more normal (Stephen and Mirman, 2010) and does not substantially affect the results of this paper.

[8]For the n-gram model, this study uses the Brown corpus (Francis and Kucera, 1979), the WSJ Sections 02-21 (Marcus et al., 1993), the written portion of the British National Corpus (BNC Consortium, 2007), and the Dundee corpus (Kennedy et al., 2003) smoothed with modified Kneser-Ney (Chen and Goodman, 1998) in SRILM (Stolcke, 2002).

[9]Non-cumulative metrics are calculated from the final word of the go-past region; cumulative metrics are summed over the go-past region.

included as predictors along with the predictors from the previous go-past region (to account for spillover effects). Finally, each model has subject and item random intercepts added in addition to by-subject random slopes (cumulative total surprisal, whether the previous word was fixated, and length of the go-past region) and is fit to centered log-transformed go-past durations.[10]

The Akaike Information Criterion (AIC) indicates that the gap-reallocating model (AIC = 128,605) provides a better fit to reading times than the original model (AIC = 128,619).[11]

As described in Section 1, previous findings of negative integration cost may be due to a confound whereby center-embedded constructions caused by modifiers, which do not require deep syntactic dependencies to be deferred, may be driving the effect. Under this hypothesis, embeddings that do not arise from final adjunction of modifiers (henceforth *canonical* embeddings) should yield a positive integration cost as found by Gibson (2000).

To investigate this potential confound, the Dundee corpus is partitioned into two parts. First, the model described in this paper is used to annotate the Dundee corpus. From this annotated corpus, all sentences are collected that contain canonical embeddings and lack modifier-induced embeddings.[12] This produces two corpora: one consisting entirely of canonical center-embeddings such as those used in self-paced reading experiments with findings of positive integration cost (e.g. Gibson 2000), the other consisting of the remainder of the Dundee corpus, which contains sentences with canonical embeddings but also includes modifier-caused embeddings.

The coefficient estimates for integration operations (–F+L and +N) on each of these corpora are then calculated using the baseline described above. To ensure embeddings are driving any observed effect rather than sentence wrap-up effects, the first and last words of each sentence are excluded from both data sets. Integration cost is measured by the amount of probability mass the parser allocates to –F+L and +N operations, accumulated over each go-past region, and this cost is added as a fixed effect and as a random slope by subject to the mixed model described earlier.[13]

Model	coeff	std err	t-score
Canonical	-0.040	0.010	-4.05
Other	-0.017	0.004	-4.20

Table 1: Fixed effect estimates for integration cost when used to fit reading times over two partitions of the Dundee corpus: one containing only canonical center embeddings and the other composed of the rest of the sentences in the corpus.

The fixed effect estimate for cumulative integration cost from fitting each corpus is shown in Table 1. Application of Welch's t-test shows that the difference between the estimated distributions of these two parameters is highly significant ($p < 0.0001$).[14] The strong negative correlation of integration cost to reading times in the purely canonical corpus suggests canonical (non-modifier) integrations contribute to the finding of negative integration cost.

6 Conclusion

This paper has introduced an incremental parser capable of using GCG dependencies to distinguish between surface syntactic embeddings and deep syntactic embeddings. This parser was shown to obtain a better fit to reading times than a surface-syntactic parser and was used to parse the Dundee eye-tracking corpus in two partitions: one consisting of canonical embeddings that require deferred dependencies and the other consisting of sentences containing no center embeddings or center embeddings arising from the attachment of clause-final modifiers, in which no dependencies are deferred. Using linear mixed effects models, completion (integration) of canonical center embeddings was found to be significantly more negatively correlated with reading times than completion of non-canonical embeddings. These results suggest that the negative integration cost observed in eye-tracking studies is at least partially due to deep syntactic dependencies and not due to confounds related to surface forms.

[10] Each fixed effect that has an absolute t-value greater than 10 when included in a random-intercepts only model is added as a random slope by-subject.

[11] The relative likelihood of the original model to the gap-sensitive model is 0.0009 (n = 151,331), which suggests the improvement is significant.

[12] Modifier-induced embeddings are found by looking for embeddings that arise from inference rules Ma-h in Section 3.

[13] Integration cost is residualized off the baseline before being centered and added as a fixed effect.

[14] Integration cost is significant as a fixed effect ($p = 0.001$) in both partitions: canonical (n = 16,174 durations) and non-canonical (n = 131,297 durations).

References

Steven P. Abney and Mark Johnson. 1991. Memory requirements and local ambiguities of parsing strategies. *J. Psycholinguistic Research*, 20(3):233–250.

R. Harald Baayen, D. J. Davidson, and Douglas M. Bates. 2008. Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, 59:390–412.

Emmon Bach. 1981. Discontinuous constituents in generalized categorial grammars. *Proceedings of the Annual Meeting of the Northeast Linguistic Society (NELS)*, 11:1–12.

Douglas Bates, Martin Maechler, and Ben Bolker, 2011. *lme4: Linear mixed-effects models using S4 classes*.

BNC Consortium. 2007. The british national corpus.

Stanley F. Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical report, Harvard University.

Vera Demberg and Frank Keller. 2008. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109(2):193–210.

W. Nelson Francis and Henry Kucera. 1979. The brown corpus: A standard corpus of present-day edited american english.

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

Edward Gibson. 2000. The dependency locality theory: A distance-based theory of linguistic complexity. In *Image, language, brain: Papers from the first mind articulation project symposium*, pages 95–126, Cambridge, MA. MIT Press.

John Hale. 2001. A probabilistic earley parser as a psycholinguistic model. In *Proceedings of the second meeting of the North American chapter of the Association for Computational Linguistics*, pages 159–166, Pittsburgh, PA.

John Hale. 2003. *Grammar, Uncertainty and Sentence Processing*. Ph.D. thesis, Cognitive Science, The Johns Hopkins University.

Philip N. Johnson-Laird. 1983. *Mental models: towards a cognitive science of language, inference, and consciousness*. Harvard University Press, Cambridge, MA, USA.

Daniel Jurafsky. 1996. A probabilistic model of lexical and syntactic access and disambiguation. *Cognitive Science: A Multidisciplinary Journal*, 20(2):137–194.

Marcel Adam Just and Patricia A. Carpenter. 1992. A capacity theory of comprehension: Individual differences in working memory. *Psychological Review*, 99:122–149.

Fred Karlsson. 2007. Constraints on multiple center-embedding of clauses. *Journal of Linguistics*, 43:365–392.

Alan Kennedy, James Pynte, and Robin Hill. 2003. The Dundee corpus. In *Proceedings of the 12th European conference on eye movement*.

Walter Kintsch. 1988. The role of knowledge in discourse comprehension: A construction-integration model. *Psychological review*, 95(2):163–182.

Nayoung Kwon, Yoonhyoung Lee, Peter C. Gordon, Robert Kluender, and Maria Polinsky. 2010. Cognitive and linguistic factors affecting subject/object asymmetry: An eye-tracking study of pre-nominal relative clauses in korean. *Language*, 86(3):561.

Richard L. Lewis and Shravan Vasishth. 2005. An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, 29(3):375–419.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Igor Mel'čuk. 1988. *Dependency syntax: theory and practice*. State University of NY Press, Albany.

Luan Nguyen, Marten van Schijndel, and William Schuler. 2012. Accurate unbounded dependency recovery using generalized categorial grammars. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING '12)*, Mumbai, India.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL'06)*.

R Development Core Team, 2010. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

William Schuler. 2009. Parsing with a bounded stack using a model-based right-corner transform. In *Proceedings of NAACL/HLT 2009*, NAACL '09, pages 344–352, Boulder, Colorado. Association for Computational Linguistics.

Damian G. Stephen and Daniel Mirman. 2010. Interactions dominate the dynamics of visual cognition. *Cognition*, 115(1):154–165.

Andreas Stolcke. 2002. Srilm – an extensible language modeling toolkit. In *Seventh International Conference on Spoken Language Processing*.

Marten van Schijndel and William Schuler. 2013. An analysis of frequency- and recency-based processing costs. In *Proceedings of NAACL-HLT 2013*. Association for Computational Linguistics.

Marten van Schijndel, Andy Exley, and William Schuler. 2012. Connectionist-inspired incremental PCFG parsing. In *Proceedings of CMCL 2012*. Association for Computational Linguistics.

Marten van Schijndel, Andy Exley, and William Schuler. in press. A model of language processing as hierarchic sequential prediction. *Topics in Cognitive Science*.

Stephen Wu, Asaf Bachrach, Carlos Cardenas, and William Schuler. 2010. Complexity metrics in an incremental right-corner parser. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)*, pages 1189–1198.