Incremental Neo-Davidsonian semantic construction for TAG

Asad Sayeed and Vera Demberg
Department of Computational Linguistics/MMCI Cluster of Excellence
Saarland University
66123 Saarbrücken, Saarland, Germany
{asayeed,vera}@coli.uni-saarland.de

Abstract

We propose a Neo-Davidsonian semantics approach as a framework for constructing a semantic interpretation simultaneously with a strictly incremental syntactic derivation using the PLTAG formalism, which supports full connectedness of all words under a single node at each point in time. This paper explains why Neo-Davidsonian semantics is particularly suitable for incremental semantic construction and outlines how the semantic construction process works. We focus also on quantifier scope, which turns out to be a particularly interesting question in the context of incremental TAG.

1 Introduction

Incremental processing formalisms have increasing importance due to the growing ubiquity of spoken dialogue systems that require understanding and generation in real-time using rich, robust semantics. Dialogue systems benefit from incremental processing in terms of shorter response time to the user's requests when the dialogue system can start interpreting and serving the request (e.g. by consulting databases, doing reference resolution, backchannelling or starting to generate an answer (Aist et al., 2007; Schuler et al., 2009; Skantze and Schlangen, 2009)) before the request is fully stated. Another use of formalisms that support strict incrementality is psycholinguistic modelling: As there is a substantial amount of evidence that human sentence processing is highly incremental, computational models of human sentence processing should be incremental to the same degree. Such models can then be used to calculate measures of human sentence processing difficulty, such as surprisal, which have been demonstrated to correspond to reading times (e.g., Levy, 2008; Mitchell et al., 2010).

Two strictly incremental versions of tree-adjoining grammar (TAG; Joshi et al., 1975) which have been proposed in recent years are DV-TAG (Mazzei et al., 2007) and PLTAG (Demberg-Winterfors, 2010). Incremental syntax is however only of limited interest without a corresponding mechanism for calculating the incremental semantic interpretation. And for that semantic model to be practically useful in psycholinguistic modelling or NLP applications such as speech recognition or dialogue systems, we believe that the semantic representation should ideally be simple, flat and usefully underspecified, in order to be used in the future in a context of compositional distributional semantics. We propose a framework in which semantic expressions are built synchronously with the syntactic tree. Simple rules are used to integrate an elementary tree's semantic expression with the semantic expression of the prefix tree at each stage. The semantic contribution of the new elementary tree is thereby added to the semantic output expression in a manner that reflects closely the order in which semantic material has arrived. The necessary semantic annotation of elementary trees can be obtained from subcategorization frame information (PropBank, FrameNet). We use a Neo-Davidsonian event-based semantics with minimal recursion.

Integrating incremental syntactic analysis with a framework of incremental semantic interpretation will allow one to model processing phenomena such as the decreased processing difficulty (1-b) (after Steedman, 2000) in comparison to

Proceedings of the 11th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+11), pages 64–72,
Paris, September 2012.

(1-a) by downranking the main verb analysis of *sent* when the subject (like *flowers*) is unlikely to fill the sender role.

(1) a. The doctor sent for the patient arrived.
b. The flowers sent for the patient arrived.

Incrementally generating the semantic interpretation requires the underspecification of the output semantics given the syntax, such as underspecifying the number of arguments of a verb or (to a greater extent than for non-incremental deviations, as we will discuss below) the scope of quantifiers.

This paper sets forth the initial proposal for this semantic formalism in terms of underlying desiderata, principles, and basic use cases. It provides one example derivation, and it outlines a way of dealing with the question of scope ambiguities, an issue which affects a number of aspects of the theoretical plausibility of a semantic formalism.

2 PLTAG Syntax

PLTAG (Demberg-Winterfors, 2010; Demberg and Keller, 2009, 2008) is a strictly incremental version of TAG. In order to achieve the strict incrementality (i.e., all words are always connected under a single root), the formalism uses *prediction trees*, which are usually not lexicalized. Each node of a prediction tree carries markers (see indices $_k$ and ^{k} in Figure 1(c)) which indicate that the predicted node has to be *verified* by a canonical (= non-predictive) TAG tree with matching structure at a later point in the derivation, in order to yield a valid derived tree.

This *verification* operation applies when an elementary tree arrives that structurally matches nodes in the prefix tree which are marked with the prediction markers. A structural match is thereby defined as the verification tree containing all nodes with identical index (i.e. all nodes that were contributed by a specific prediction tree). Additionally, the verification tree can have further nodes to the right of the spine (= the path from the root to the anchor). The prediction markers are removed, and the lexical item in the elementary tree is placed in the head of the prediction tree as in Figure 4, and any additional nodes of the verification tree which were not part of the prediction tree are inserted into the prefix tree. For more details see Demberg-Winterfors (2010).

3 Semantic PLTAG

Consider the following ways of stating a command to a hypothetical restaurant reservation system:

(2) a. Send every restaurant a reservation request.
b. Send a reservation request to every restaurant.

A system that derives syntax and semantics incrementally and simultaneously can partially disambiguate the ambiguity between the two elementary trees of "send" (see Figure 1(a)) by taking into account that restaurants are better recipients than reservation requests, while reservation requests are more typically sent. The following sections outline how we decorate the syntactic PLTAG trees with semantic annotations, how semantic expressions are composed, and how we deal with quantifier scope.

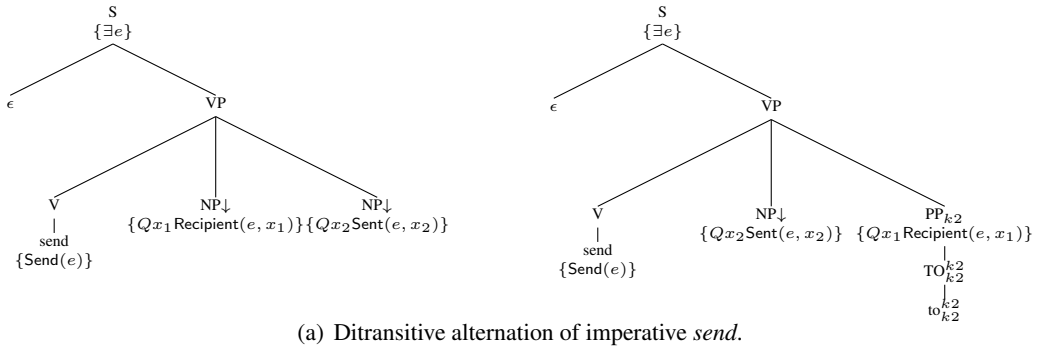
4 Neo-Davidsonian semantics

Neo-Davidsonian semantics (Parsons, 1990) is a form of first-order logic that uses existentially-bound event variables ($\exists e$) to connect verb predicates and their subcategorized arguments and separates predicate arguments into their own, separate event-modifying predicates connected through conjunctions. This permits a flexible means to underspecify function composition and argument structure (Hunter, 2009) by greatly limiting recursion (Example (3-a)) in semantic expressions.

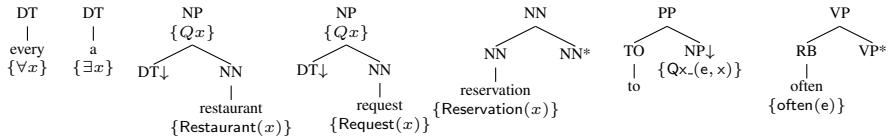
(3) a. $\mathsf{Happily}(\mathsf{Eating}(\mathsf{Candy}))$
b. $\exists x_1 \exists e \mathsf{Candy}(x_1) \& \mathsf{Eaten}(e, x_1) \& \mathsf{Eating}(e) \& \mathsf{Happily}(e)$

The Neo-Davidsonian approach has been implemented in formalisms such as Robust Minimal Recursion Semantics (RMRS; Copestake, 2007). We use a variant exemplified in (3-b).

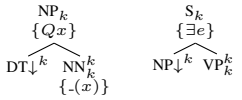
Neo-Davidsonian semantics has some advantages in the case of a strictly incremental parsing process: in incremental PLTAG parsing, the prediction trees do not always specify the full subcategorization frame of a predicate. For example, when processing the words *Peter often*, the NP tree for *Peter* and the auxiliary tree for *often* (see Fig. 1(b)) will be connected with a prediction



(a) Ditransitive alternation of imperative *send*.



(b) Canonical elementary trees for nominals, determiners, to-PP, adverbial.



(c) Prediction trees

Figure 1: Example Lexicon

tree, expressing that a verb phrase is expected and that the structure is going to root in S, see right hand tree in Fig. 1(c). It does however not specify the subcategorization frame of the verb that is going to be the head of the predicted verb phrase. Neo-Davidsonian semantics allows us to keep the same level of underspecification in the semantics, such that the verb-phrase prediction tree only introduces the event variable $\exists e$, which is then available to be unified with the unbound variable of *often*. Furthermore, the breaking up of n-ary predicates into an event predicate and a binary relation for each thematic role in Neo-Davidsonian semantics allows us to calculate the semantic contribution of a verb's argument before having seen all arguments.

The minimally-recursive nature of the formalism also permits the order of predicates in the formalism to reflect approximately the order in which their corresponding syntactic fragments were incorporated into the structure. In our PLTAG-based formalism, we tie each elementary tree to a neo-Davidsonian expression fragment which will be appended to the end of a semantic structure that grows along with the parse tree. This incorporates a notion of recency directly into our semantic expression construction, a characteristic relevant to semantic enhancements of PLTAG's syntactic prediction component.

This enables the produced expressions to be used directly in statistical prediction techniques where order may matter, including compositional distributional semantics (Mitchell and Lapata, 2010) or HMM-style sequence learning techniques. Its trade-off is that the semantic expressions are not always guaranteed to be immediately well-formed, particularly in the order of scopes. However, in the later sections we discuss ways to identify structure that needs to be rearranged, ways that can be applied at every incremental step.

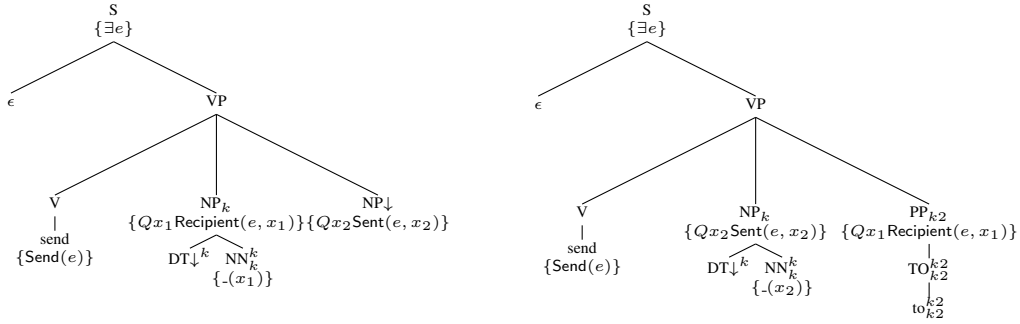
4.1 Lexical construction and derivation

These are some relevant aspects of the lexicon's construction:

Verb trees are annotated at the root with an existentially-quantified event variable. The head node contains the verb's own predicate, and the nodes representing arguments have argument predicates with entity variables. The argument predicates correspond to thematic roles.

The semantic expressions associated with elementary trees consist of four types of variables, entity variables, written as x_n, first-order predicate variables, written as $\_$, quantifier variables, written as Q_n, and event variables, written as e_n. $\_$ variables are associated with the predictive component of PLTAG-based parsing and achieve their

Sentence: *Send || every restaurant a reservation request*



Semantics: $\exists e Q x_1 \mathsf{Recipient}(e, x_1) \& \_(x_1) \& Q x_2 \mathsf{Sent}(e, x_2)$ $\& \mathsf{Send}(e)$

Semantics: $\exists e Q x_2 \mathsf{Sent}(e, x_2) \& \_(x_2) \& Q x_2 \mathsf{Recipient}(e, x_1)$ $\& \mathsf{Send}(e)$

Figure 2: Ambiguous derivation when ditransitive *Send* is received.

values whenever a verification occurs that unifies a prediction tree with an elementary tree. Q_n variables are given concrete quantifier values whenever an elementary tree with a quantifier is substituted into a noun elementary tree that has the Q_n variable. e_n variables represent events and their connections to event participants through role predicates.

Determiners represent their own appropriate quantifier (e.g. $\forall x$). NP-rooted elementary trees for nouns have an unspecified Q quantifier at the root and a semantic predicate corresponding to the noun. Nominal adjuncts (such as the example *reservation* auxiliary tree) contain a corresponding predicate over an unbound entity variable.

4.2 Composition rules

We describe our procedure for incremental semantic parsing in terms of triggered procedures for the incorporation of the semantic expressions residing on nodes of the most recently attached elementary tree. We first describe some common characteristics of the composition procedure, and then we describe how some frequent types of elementary trees are semantically processed on arrival. One aspect of our approach is that most of the "work" in building the semantic expression is defined in terms of the derived tree, in contrast to other TAG semantics approaches that use the derivation tree. We found this the simplest way to align the verification step's role in giving values to prediction trees and the valuation of _ variables in the semantic expressions. Expressions and variables in the derived tree are indexed to their corresponding forms in the semantic expression, so that they grow in parallel.

The semantic component proceeds through rules for unification and predicate emission. Semantic predicates are emitted as conjuncts from left to right, normally revising previous semantic structure based on unification events during PLTAG operations. Whenever an elementary tree is either substituted, adjoined, or predicted, the semantic expressions on that tree's nodes are emitted as conjuncts, once all unifications have been resolved.[1]

When an unbound variable joins the structure, it is unified with the nearest compatible variable the shortest distance above it in the structure. (Unbound variables generally appear in elementary trees representing adjunct structures like adjectives.) For the x_n entity variables, when they are unified, they will be assigned the same variable subscript. For Q quantifier variables, they will search for the nearest quantifier or quantifier variable, by the same standard of nearness. _ variables are predictive and only bound during verification.

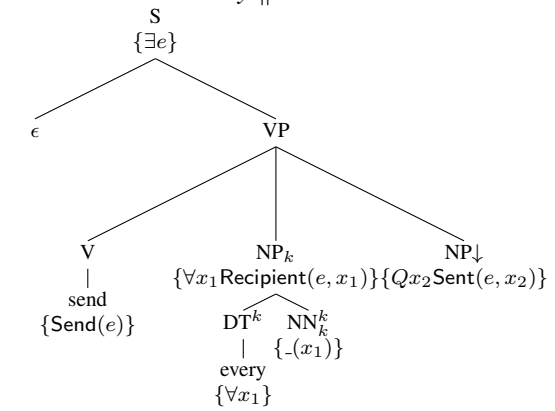
By default, all predicates are joined with conjunction operators, except in the case where a $\forall x_n$ is called for. Then a $\rightarrow$ conditional operator must be inserted when the universally-quantified NP is complete, with all adjuncts in the restrictor of the quantifier. We describe later a procedure to identify NP adjuncts directly from the semantic expression in order to insert the conditional operator as needed.

4.2.1 Example derivation

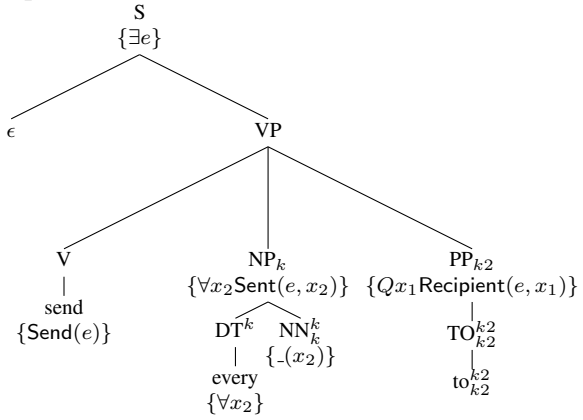
Figures 2-6 represent an example derivation based on the sentence *Send every restaurant a reservation request* using the lexicon in figure 1.

[1] We use the word "unification" in the sense of establishing strict structural identity and variable coindexation as in the Prolog programming language.

Sentence: *Send every* ‖ *restaurant a reservation request*



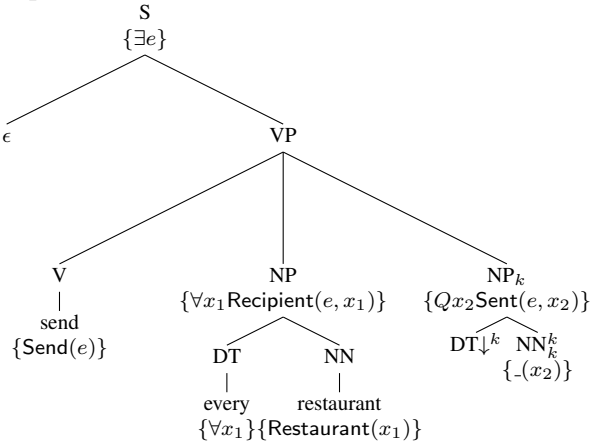
Semantics: $\exists e \forall x_1 \mathsf{Recipient}(e, x_1) \& \_(x_1) \& Q x_2 \mathsf{Sent}(e, x_2) \; \& \mathsf{Send}(e)$



Semantics: $\exists e \forall x_2 \mathsf{Sent}(e, x_2) \& \_(x_2) \& Q x_2 \mathsf{Recipient}(e, x_1) \; \& \mathsf{Send}(e)$

Figure 3: Derivation remains ambiguous after *every* is received.

Sentence: *Send every restaurant* ‖ *a reservation request*



Semantics: $\exists e \forall x_1 \mathsf{Recipient}(e, x_1) \& \mathsf{Restaurant}(x_1) \; \& Q x_2 \mathsf{Sent}(e, x_2) \& \mathsf{Send}(e) \& \_(x_2)$

Figure 4: The prediction NP tree is verified through unification with the tree for *restaurant*.

In figure 2, the two possible trees for *send* are introduced from the lexicon, and the semantic expressions associated with the nodes are emitted.

In figure 3, the derivation remains ambiguous after *every* is received. However, the quantifier $\forall$ unifies with the quantifier variable represented as Q in both the trees and in the resulting semantics.

Next, the prediction NP tree is verified through unification with the tree for *restaurant* (figure 4). The empty predicate _ is also filled. Semantic selectional constraints abolish the second parallel derivation. The derivation is also now ready for the next NP prediction tree.

Prediction of a determiner and verification with *a* proceeds in analogy to the first NP. We then adjoin the noun *reservation* (figure 5). For brevity, we omit the ambiguity created by the possible interpretation of *reservation* as an argument; we are illustrating the effect of adjunction. The variable in $\mathsf{Reservation}(x)$ unifies with the nearest quantified variable along the spine of the tree; nominal adjuncts generally do not bring entity variable bindings with them. The predicate is literally appended to the semantic expression as another conjunct.

Finally (figure 6), the prediction tree is verified by the arrival of *request*, which fills out the last _ predicate variable. We demonstrate in this step an adjustment of the semantics to a form in which the event is quantified in the lowest position and the restrictor of $\forall$ is correctly placed before an inserted $\rightarrow$. We outline how to make this work in the next section.

4.3 Adjuncts and arguments

It is sometimes important to identify the parts of the output semantic expression that pertain to adjuncts, especially when interpreting the positions of variable scope bindings and satisfying the semantic conditions thereon. The order of strict incremental appearance of predicates and variables may be subject to further interpretive conditions that require limited reordering of sub-expressions, depending on the application. For example, Champollion (2011) notes that existentially-quantified events should be interpreted in the lowest possible position relative to the bindings of the event's arguments. This requires some ability to distinguish between argument and adjunct predicates. Similarly, in order to handle scope ambiguities soundly, the system

must also have the capacity to distinguish between the restrictor and the nuclear scope of a quantifier.

There are multiple ways to do this, including from the structure of the derived tree and the order in which the derivation proceeded, but we argue that most of the work can be done within the semantic expression itself. Most trivially, nominal adjuncts can be identified by the lack of an event argument to their predicates. Relative clause adjuncts will contain their own event variables, but will not refer to the exterior event directly. As an illustration:

(4) a. Some flower that some bride holds wilts.
b. $\exists x_1 \mathsf{Flower}(x_1) \& [\exists x_2 \mathsf{Bride}(x_2)$
$\& \exists e_2 \mathsf{Hold}(e_2) \& \mathsf{Holder}(e_2, x_2)$
$\& \mathsf{Held}(e_2, x_1)] \& \exists e_1 \mathsf{Wilt}(e_1) \& \mathsf{Wilter}(e_1, x_1)$

Since e_1 is the root event, we know that the relative clause "that the bride holds" does not directly refer to it in its semantic expression and is an adjunct only of "the flower". We can even deduce from the expression that "the flower" is the host NP for the relative clause, because "the bride" is not directly connected to the root event as an argument. We can therefore correctly insert the $\rightarrow$ if the matrix subject had a universal quantifier.

(5) a. Every flower that some bride holds wilts.
b. $\forall x_1 \mathsf{Flower}(x_1) \& [\exists x_2 \mathsf{Bride}(x_2)$
$\& \exists e_2 \mathsf{Hold}(e_2) \& \mathsf{Holder}(e_2, x_2) \& \mathsf{Held}(e_2, x_1)]$
$\rightarrow \exists e_1 \mathsf{Wilt}(e_1) \& \mathsf{Wilter}(e_1, x_1)$

A similar procedure can be applied to the distinction between argument clauses that appear after verbs like "say" and adjunct clauses heralded by "because." Argument clauses are heralded by an event variable that itself becomes an argument of a role predicate of the matrix argument, as here:

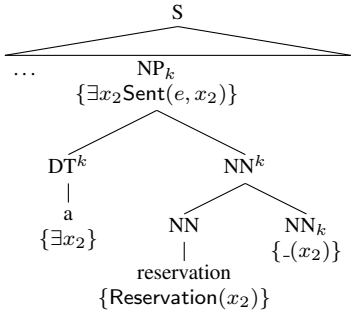
(6) a. Some professor says some student failed.
b. $\exists x_1 \mathsf{Professor}(x_1) \& \exists e_1 \mathsf{Say}(e_1)$
$\& \mathsf{Speaker}(e_1, x_1) \& [\exists x_2 \mathsf{Student}(x_2)$
$\& \exists e_2 \mathsf{Spoken}(e_1, e_2) \& \mathsf{Fail}(e_2) \& \mathsf{Failer}(e_2, x_2)]$

In the case of an adjunct clause introduced with "because", there is no role predicate connecting the subordinate event to the matrix event, which is practically the definition of an adjunct.

4.4 Scope and Underspecification

Our example derivation in Figures 2 to 6 shows how a syntactic tree and semantic interpretation

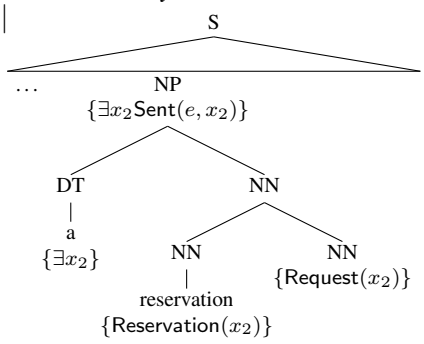
Sentence: *Send every restaurant a reservation* ‖ *request*



Semantics: $\exists e \forall x_1 \mathsf{Recipient}(e, x_1) \& \mathsf{Restaurant}(x_1)$
$\& \exists x_2 \mathsf{Sent}(e, x_2) \& \mathsf{Send}(e) \& \_(x_2) \& \mathsf{Reservation}(x_2)$

Figure 5: Prediction of a determiner and verification with *a* proceeds in analogy to the first NP. We then adjoin the noun *reservation*.

Sentence: *Send every restaurant a reservation request* ‖



Semantics: $\forall x_1 \mathsf{Restaurant}(x_1) \rightarrow \exists x_2 \mathsf{Request}(x_2)$
$\& \mathsf{Reservation}(x_2) \& \exists e \mathsf{Recipient}(e, x_1)$
$\mathsf{Sent}(e, x_2) \& \mathsf{Send}(e)$

Figure 6: Finally, the prediction tree is verified by the arrival of *request*, which fills out the last _ predicate variable.

can be constructed for the sentences in (2-a). However, it only derives one possible reading for the sentence $\exists > \forall$, i.e. there exists a single reservation request, and that it is to be sent indiscriminately to all restaurants. This reading reflects the linear order that quantifiers occurred in. It misses the other interpretation $\forall > \exists$ (which was likely intended by the user). When sentences are derived incrementally, we cannot choose in the derivation to first substitute the second NP in order to get the other reading. Therefore, we need to systematically use *underspecification* to get both readings. Note that this problem also exists to a certain extent in standard LTAG (Joshi et al., 2007; Barker, 2010).

However, not all quantifier orderings are permissible given the syntax. Joshi et al. (2007) and Romero and Kallmeyer (2005) present an example of what can be seen as a challenge for current TAG-based and movementless approaches:

(7) Two politicians spied on some person from every city.

They describe a situation where syntactic constraints prevent some readings out of all possible permutations of quantifiers. In order to forbid their hypothetical widest matrix scope reading of *every city*, we use dominance constraints (Koller et al., 2003) to implement restrictions based on the argument-adjunct distinction in preventing quantifier-raising.

Ruys and Winter (2010) describe two major approaches to developing a formal treatment of scope ambiguities. One of these approaches is the covert movement approach that comes from the Chomskyan generative tradition, sometimes known as Quantifier Raising (QR). In this approach, alternative scope readings are found by applying highly constrained operators over an intermediate representation. The other approach involves directly embedding the mechanism for the observed scope readings in the logical representation of the semantics, typically assisted by type-raising operators.

The QR approach stems from the observation that there appears to be a close relationship between *wh*-movement islands and the restrictions on scope readings, with *wh*-islands partially exhibited through restrictions on overt movement as well as covert movement. In other words, where an inverse scope reading is impossible, it is the case in languages that are not *wh in situ* that the overt *wh*-movement is also generally difficult or impossible.

This insight is difficult to capture in highly lexicalized, movement-less formalisms. Since *wh*-movement is overt, it is possible to lexicalize these types of structures and analyse them in a movement-less way, mostly from the surface structure. Quantifier scope ambiguity, however, "lives after the syntax" in some sense: further stipulations must be made in order to enable readings that do not come directly from the surface order. Multi-component TAG (MCTAG) formalisms achieve this by permitting TAG structures with ambiguous syntactic attachments for quantifiers; then it is possible to achieve scope ambiguity through the underspecification of the position of the quantifiers in the semantics (Joshi et al., 2007).

If, on the other hand, we need an explicit representation of the quantifier positions in syntax and semantics—for example, at every step in statistically-guided predictive parsing—then we need also to make the operations that convert semantic expressions from surface to inverse scope readings a little more explicit. Champollion (2011) presents a means of enabling quantifier raising in a Neo-Davidsonian semantic formalism without movement by optionally applying a type-shifting operator to quantified items. The change in type is propagated up the syntactic tree via lambda-calculus operations in order to provide an expression with the intended scope.

However, the potential bottom-up recalculation of the entire semantic expression is also not particularly friendly to a parsing technique that is striving to be meaningfully incremental and predictive. Instead, we propose to bring back, in a very limited fashion, a movement-based analysis of QR.

For the object noun phrase in example (7), both scopal readings are possible: there was a single person whose origin at some point has been every city ($\exists > \forall$), or for every city, there is a person from that city ($\forall < \exists$)—they are interchangeable. For the whole sentence, one of several available readings is that for every city, there is a person whom two politicians spied on ($\forall > \exists < 2$). However, a reading that is *not* available is $*\forall > 2 > \exists$—that every city has its own pair of politicians who are spying on some person in that city.[2]

[2]We can find a parallel for this distinction in *wh*-movement islands. Consider the following question, analogous to the $*\forall > 2 > \exists$:

(i) *From which cities did two politicians spy on some person? (with the interpretation that "from which cities" semantically applies to "some person").

Adjunct phrases are islands for *wh*-movement, leading to the ungrammaticality of the reading. However, it is possible to ask a highly emphatic multiple question:

(ii) Which people from which cities did two politicians spy on?

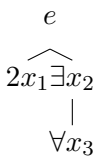
It is possible to bring the adjunct *wh* along with the argument *wh* in this case of overt movement. In fact, many languages (Bošković, 2002) permit multiple *wh*-movement, constrained only by island restrictions such as adjunct islands. Multiple *wh*-movement by pied-piping is permitted in English when the lower question is a syntactic adjunct (Reich, 2002).

Furthermore, Romero and Kallmeyer (2005) present an

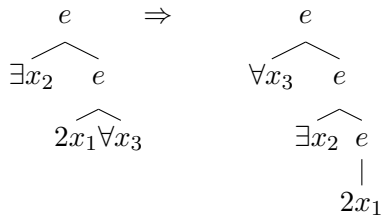
Focusing specifically on relations between quantifiers, we can define a representation and a set of constraints that limit the possible readings by looking at the relationships between variables in our semantic expression formalism. We construct a minimal spanning tree of variables connected to the event variable of the predicate in the main clause, which is then the root of the tree. Each edge in the variable tree, which is roughly analogous to a very stripped-down TAG derivation tree, represents a pair of arguments present in a binary predicate in the expression; parent-child relationships between event variables and entity variables are only permitted through role predicates to prevent the events of adjunct clauses from participating in scope relations. Each x_n variable would be annotated in the tree by its associated quantifier. For example, consider the following interpretation of example (7) in our formalism, after the lowering of the event scope and insertion of $\rightarrow$:

(8) $2x_1\mathsf{Politician}(x_1)\&\exists x_2\mathsf{Person}(x_2)\&\forall x_3$
$\mathsf{City}(x_3)\&\mathsf{From}(x_2, x_3) \rightarrow$
$\exists e\mathsf{Spyer}(e, x_1)\&\mathsf{SpiedUpon}(e, x_2)\&\mathsf{Spy}(e)$

The variable tree for this would be:

(9)


We interpret surface scope order left to right and top down. Then we can express the interchangeability of $\exists$ and $\forall$ by declaring that variables in a parent-child relationship can be interpreted interchangeably. Another important stipulation over variable trees is that only the immediate children of an event variable can change their relationship to the other descendants of the event variable, which would rule out $\forall$ immediately taking scope over 2 and $\exists$. This represents adjunct island restrictions on covert movement. The $\forall > \exists > 2$ reading can be licensed by actually allowing movement to occur in a manner somewhat analogous to the move operator in Minimalist accounts with no traces.

(10)


The last step is permissible because $\forall x_3$ is an immediate child of e in the previous step. Extending this analysis to, e.g., forbid QR from causing subjacency violations can be accomplished in approximately the following way: no entity variable can rise above its own nearest ancestor event variable except in the circumstance that the event variable is itself a child of another event variable (equivalent to successive-cyclic movement). These scope order changes can be reflected in the semantics, if necessary, by explicitly reordering the affected variable bindings in the expression. We leave a fuller exploration of the details of this proposal to future work.

5 Concluding remarks

We have proposed here a semantic extension to PLTAG, a syntactic formalism intended to enable robust, psychologically-plausible incremental parsing. This presented us the challenge of reconciling potentially conflicting goals, such as strict incrementality, semantic well-formedness, psycholinguistic plausiblity, and engineering applicability. Our solution consists of a variant of neo-Davidsonian semantics adjusted to support a close synchronisation between the composition of output semantic expressions and PLTAG operations such as the prediction/verification mechanism.

We illustrated the formalism with an example parse, and we described a way in which our ap-

additional non-surface reading, $\exists > \forall, 2$, where $\forall$ is in the restrictor of $\exists$ (not surprising due to the adjunct status of "from every city"). While the requirement for pied-piping in English requires both *wh*-phrases to be fronted, this is a restriction that belongs to the syntax. Many analyses show that pied-piping restrictions do not necessarily hold at the level of logical form (von Stechow, 1996; Reich, 2002), even if island constraints do. Consequently, the additional valid reading is not ruled out by a QR analysis based in covert movement.

Given this parallelism, one could reasonably be tempted to advocate for the re-adoption of movement-based analyses in movementless grammatical formalisms, but at least in the case of TAG, actually doing so would break many of the properties of the formalism. Consequently, we only propose a form of movement-equivalent operation that focuses on representing ambiguities in semantic structure that are not easily accommodated in a monotonic, movementless incremental parsing system.

proach can be adapted to handle ambiguous scope phenomena that constitute a challenge for robust semantic representation. Potential extensions of this work include semantic PLTAG lexicon extraction from treebanks and further formalisation of our variable tree representation of scope phenomena.

References

Aist, G., Allen, J., Campana, E., Gallo, C., Stoness, S., Swift, M., and Tanenhaus, M. (2007). Incremental dialogue system faster than and preferred to its nonincremental counterpart. pages 761–766.

Barker, C. (2010). Cosubstitution, derivational locality, and quantifier scope.

Bošković, Ž. (2002). On multiple wh-fronting. *Linguistic inquiry*, 33:351–383.

Champollion, L. (2011). Quantification and negation in event semantics. *The Baltic international yearbook of cognition, logic, and communication*, 6(0).

Copestake, A. (2007). Semantic composition with (robust) minimal recursion semantics. In *Proc. of the Workshop on Deep Linguistic Processing*.

Demberg, V. and Keller, F. (2008). A psycholinguistically motivated version of TAG. In *Proc. of the 9th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+9)*, Tübingen, Germany.

Demberg, V. and Keller, F. (2009). A computational model of prediction in human parsing: Unifying locality and surprisal effects.

Demberg-Winterfors, V. (2010). *A broad-coverage model of prediction in human sentence processing*. PhD thesis, School of Informatics, The University of Edinburgh.

Hunter, T. (2009). Deriving syntactic properties of arguments and adjuncts from neo-davidsonian semantics. In *Proc. of MOL 2009*, Los Angeles, CA, USA.

Joshi, A., Kallmeyer, L., and Romero, M. (2007). Flexible composition in ltag: quantifier scope and inverse linking. *Computing meaning*, pages 233–256.

Joshi, A., Levy, L., and Takahashi, M. (1975). Tree adjunct grammars. *Journal of the Computer and System Sciences*, 10.

Koller, A., Niehren, J., and Thater, S. (2003). Bridging the gap between underspecification formalisms: Hole semantics as dominance constraints. In *Proc. of EACL 2003*, pages 367–374.

Levy, R. (2008). Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Mazzei, A., Lombardo, V., and Sturt, P. (2007). Dynamic tag and lexical dependencies. *Research on Language and Computation, Foundations of Natural Language Grammar*, pages 309–332.

Mitchell, J. and Lapata, M. (2010). Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

Mitchell, J., Lapata, M., Demberg, V., and Keller, F. (2010). Syntactic and semantic factors in processing difficulty: An integrated measure. In *Proc. of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 196–206, Uppsala, Sweden.

Parsons, T. (1990). *Events in the semantics of English*. MIT Press, Cambridge, MA, USA.

Reich, I. (2002). Pied piping and the syntax and semantics of wh-phrases. In Mauck, S. and Mittelstaedt, J., editors, *Georgetown University working papers in theoretical linguistics*, volume 2, pages 263–286.

Romero, M. and Kallmeyer, L. (2005). Scope and situation binding in ltag using semantic unification. In *Proc. of the 7th international workshop on computational semantics (IWCS)*.

Ruys, E. and Winter, Y. (2010). Scope ambiguities in formal syntax and semantics. In Gabbay, D. and Guenthner, F., editors, *Handbook of Philosophical Logic*, volume 16, pages 159–225.

Schuler, W., Wu, S., and Schwartz, L. (2009). A framework for fast incremental interpretation during speech decoding. *Computational Linguistics*, 35(3):313–343.

Skantze, G. and Schlangen, D. (2009). Incremental dialogue processing in a micro-domain. In *Proc. of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 745–753.

Steedman, M. (2000). *The syntactic process*. MIT Press.

von Stechow, A. (1996). Against LF pied-piping. *Natural Language Semantics*, 4(1):57–110.