The semantic augmentation of a psycholinguistically-motivated syntactic formalism

Asad Sayeed and Vera Demberg
Computational Linguistics and Phonetics / M²CI Cluster of Excellence
Saarland University
66123 Saarbrücken, Germany
`{asayeed,vera}@coli.uni-saarland.de`

Abstract

We augment an existing TAG-based incremental syntactic formalism, PLTAG, with a semantic component designed to support the simultaneous modeling effects of thematic fit as well as syntactic and semantic predictions. PLTAG is a psycholinguistically-motivated formalism which extends the standard TAG operations with a prediction and verification mechanism and has experimental support as a model of syntactic processing difficulty. We focus on the problem of formally modelling semantic role prediction in the context of an incremental parse and describe a flexible neo-Davidsonian formalism and composition procedure to accompany a PLTAG parse. To this end, we also provide a means of augmenting the PLTAG lexicon with semantic annotation. To illustrate this, we run through an experimentally-relevant model case, wherein the resolution of semantic role ambiguities influences the resolution of syntactic ambiguities and vice versa.

1 Introduction

PLTAG (PsychoLinguistically-motivated TAG, Demberg and Keller, 2008; Demberg et al., 2014) is a variant of Tree-Adjoining Grammar (TAG) which is designed to allow the construction of TAG parsers that enforce strict incrementality and full connectedness through (1) constraints on the order of operations, (2) a new type of unlexicalized tree, so-called prediction trees, and (3) a verification mechanism that matches up and extends predicted structures with later evidence. Psycholinguistic evaluation has shown that PLTAG operations can be used to predict data from eye-tracking experiments, lending this syntactic formalism greater psycholinguistic support.

Syntax, however, may not just be the skeleton of a linguistic construction that bears semantic content: there is some evidence that syntactic structure and semantic plausibility interact with each other. In a strongly interactive view, we would expect that semantic plausibility could directly affect the syntactic expectations. Consider the sentences:

(1) a. The woman slid the butter to the man.
    b. The woman slid the man the butter.

The ditransitive verb "to slide" provides three roles for participants in the predicate: agent, patient, and recipient. In both cases, "the woman" fills the agent role, "the butter" the patient, and "the man" the recipient. However, they do not generally fill all roles equally well. English-speakers have the intuition that "the butter" should neither be an agent nor a recipient under normal circumstances. Likewise, "the man" is not a typical patient in this situation. If there is a psycholinguistic effect of semantic plausibility, we would expect that an incomplete sentence like "The woman slid the butter" would generate an expectation in the listener of a PO construction (rather than DO) with preposition "to", as well as an expectation of a noun phrase and an expectation that that noun phrase would belong to the class of entities that are plausible recipients for entities that are slid.

If this is the case, then there is not only a syntactic expectation at this point but a semantic expectation that is in turn informed by the syntactic structure and semantic content up to that point. Constructing a model that is formally rich, psycholinguistically plausible, and empirically robust requires making design decisions about the specific relationship between syntax and semantics and the overall level of formal articulation on which the statistical model rests. For PLTAG, we are interested in preserving as many of its syntactic characteristics as are necessary to model the phenomena that it already does (Demberg and Keller, 2009).

57

In the rest of this paper, we therefore present a semantic augmentation of PLTAG that is based on neo-Davidsonian event semantics and is capable of supporting incrementality and prediction.

2 Psycholinguistic background

Does thematic fit dynamically influence the choice of preferred syntactic structures, does it shape predictions of upcoming semantic sorts, and can we measure this experimentally?

A classic study (Altmann and Kamide, 1999) about the influence of thematic fit on predictions showed that listeners can predict the complement of a verb based on its selectional restrictions. Participants heard sentences such as:

(2) a. The boy will *eat* the cake.
    b. The boy will *move* the cake.

while viewing images that depicted sets of relevant objects, in this example, a cake, a train set, a ball, and a model car. Altmann and Kamide (1999) monitored participants' eye-movements while they heard the sentences and found an increased number of looks to the cake during the word *eat* compared the control condition, i.e., during the word *move* (only the cake is edible, but all depicted objects are movable). This indicates that selectional preference information provided by the verb is not only used as soon as it is available (i.e., incremental processing takes place), but this information also triggers the prediction of upcoming arguments of the verb. Subsequent work has demonstrated that this is not a simple association effect of *eat* and the edible item *cake*, but that people assign syntactic roles rapidly based on case marking and that missing obligatory thematic role fillers are predicted; in a German visual world study, Kamide et al. (2003a) presented participants with a scene containing a cabbage, a hare, a fox and a distractor object while they heard sentences like

(3) a. Der Hase frisst gleich den Kohl.
       (*The hare$_{nom}$ will eat soon the cabbage$_{acc}$.*)
    b. Den Hasen frisst gleich der Fuchs."
       (*The hare$_{acc}$ will eat soon the fox$_{nom}$.*)

They found that, during the verb-adverb region, people looked more to the cabbage in the first condition and correctly anticipated the fox in the second condition. This means that they were able to correctly anticipate the filler of the missing thematic role. Kamide et al. (2003b) furthermore showed that role prediction is not only restricted to the immediately-following grammatical object, but that goals as in *The woman slid the butter to the man* are also anticipated.

Thematic fit furthermore seems to interact with syntactic structure. Consider the sentences in (4), which are locally ambiguous with respect to a main clause interpretation or a reduced relative clause.

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

Comprehenders incur decreased processing difficulty in sentences like (4-b) compared to (4-a), due to flowers not being a good thematic fit for the agent role of sending (Steedman, 2000).

Taken together, the experimental evidence suggests that semantic information in the form of thematic fit can influence the syntactic structures maintained by the comprehender and that people do generate anticipations not only based on the syntactic requirements of a sentence, but also in terms of thematic roles. While there is evidence that both syntactic and semantic processing is rapid and incremental, there remain, however, some open questions on how closely syntactic and semantic processing are integrated with each other. The architecture suggested here models the parallel, highly incremental construction of syntactic and semantic structure, but leaves open to exploration the question of how quickly and strongly they interact with each other. Note that with the present architecture, thematic fit would only be calculated for word pairs which stand in a possible syntactic relation. The syntax thus exerts strong constraints on which plausibilities are considered. Our example in section 6.2 illustrates how even a tight form of direct interaction between syntax and semantics can be modelled.

3 Relation to previous work on joint syntactic-semantic models

Previous attempts have been made to combine the likelihood of syntactic structure and semantic plausibility estimates into one model for predicting human processing difficulty (Padó et al., 2009; Jurafsky, 2002). Padó et al. (2009) predict increased difficulty when the preferred syntactic analysis is incompatible with the analysis that would have the best thematic fit. They integrate syntactic and semantic models as a weighted combination of plausibility scores. The syntactic

and semantic models are computed to some extent independently of one another, and then the result is adjusted by a set of functions that take into account conflicts between the models. In relation to the approach proposed here, it is also important to note that the semantic components in (Padó et al., 2009; Jurafsky, 2002) are limited to semantic role information, while the architecture proposed in this paper can build complete semantic expressions for a sentence. Furthermore, these models do not model the prediction and verification process (in particular, they do not make any semantic role predictions of upcoming input) which has been observed in human language processing.

Mitchell et al. (2010) propose an integrated measure of syntactic and semantic surprisal as a model of processing difficulty, and show that the semantic component improves modelling results over a syntax-only model. However, the syntactic and semantic surprisal components are only very loosely integrated with one another, as the semantic model is a distributional bag-of-words model which does not take syntax into account.

Finally, the syntactic model underlying (Padó et al., 2009; Mitchell et al., 2010) is an incremental top-down PCFG parser (Roark, 2001), which due to its parsing strategy fails to predict human processing difficulty that arises in certain cases, such as for center embedding (Thompson et al., 1991; Resnik, 1992). Using the PLTAG parsing model is thus more psycholinguistically adequate.

3.1 Towards a broad-coverage integration of syntax and semantics

The current paper does not propose a new model of sentence processing difficulty, but rather explores the formal architecture and mechanism necessary to enable the future implementation of an integrated syntactic-semantic model. A syntax-informed semantic surprisal component implemented using distributional semantics could use the semantic expressions generated during the PLTAG semantics construction to determine what words (in which relationships to the current word) from the previous context to condition on for calculating semantic surprisal.

4 PLTAG syntax

PLTAG uses the standard operations of TAG: substitution and adjunction. The order in which they are applied during a parse is constrained by in-

crementality. This also implies that, in addition to the standard operations, there are reverse Up versions of these operations where the prefix tree is substituted or adjoined into a new elementary tree (see figure 4). In order to achieve strict incrementality and full connectedness at the same time while still using linguistically motivated elementary trees, PLTAG has an additional type of (usually) unlexicalized elementary tree called prediction trees. Each node in a prediction tree is marked with upper and/or lower indices $_k^k$ to indicate its predictive status. Examples for prediction trees are given at the right hand side of figure 5b. The availability of prediction trees enable a sentence starting with "The thief quickly" to integrate both the NP ("The thief") and the ADVP ("quickly") into the derivation even though neither type of elementary tree can be substituted or adjoined to the other—the system predicts an S tree to which both can be attached, but no specific verb head. Prediction markers can be removed from nodes via the verification operation, which makes sure that predicted structure is matched against actually observed evidence from the input string. For the example above, the verb *ran* in "The thief quickly ran" verifies the predicted verb structure. In figures 5c through 5e, we also provide an example of prediction and verification as part of the demonstration of our semantic framework. Other foundational work on PLTAG (Demberg-Winterfors, 2010) contains more detailed description.

5 Neo-Davidsonian semantics

Davidsonian semantics organizes the representation of predicates around existentially-quantified event variables (e). Sentences are therefore treated as descriptions of these events, leading to a less recursive representation where predicates are not deeply embedded inside one another. Highly recursive representations can be incrementality-unfriendly, potentially requiring complex inference rules to "undo" recursive structures if relevant information arrives later in the sentence.

Neo-Davidsonian semantics (Parsons, 1990; Hunter, 2009) is an extension of Davidsonian semantics wherein the semantic roles are also separated out into their own first-order predicates, rather than being fixed arguments of the main predicate of the verb. This enables a single verb predicate to correspond to multiple possible arrangements of role predicates, also an

incrementality-friendly characteristic[1]. The Neo-Davidsonian representation allows us separate the semantic prediction of a role from its syntactic fulfillment, permitting the type of flexible framework we are proposing in this paper.

We adopt a neo-Davidsonian approach to semantics by a formalism that bears similarity to existing frameworks such as (R)MRS (Robust Minimal Recursion Semantics) (Copestake, 2007). However, this paper is intended to explore what architecture is minimally required to augment the PLTAG syntactic framework, so we do not adopt these existing frameworks wholesale. Our examples such as figures 4, 5d, and several others demonstrate how this looks in practice.

6 Semantics for PLTAG

6.1 Semantic augmentation for the lexicon

Constructing the lexicon for a semantically augmented PLTAG uses a process based on the one for "purely syntactic" PLTAG. The PLTAG lexicon is extracted automatically from the PLTAG treebank, which has been derived from the Penn Treebank using heuristics for binarizing flat structures as well as additional noun phrase annotations (Vadas and Curran, 2007), PropBank (Palmer et al., 2003), and a slightly modified version of the head percolation table of Magerman (1994). PLTAG trees in the treebank are annotated with syntactic headedness information as well as information that allows one to distinguish arguments and modifiers.

Given the PLTAG treebank, we extract the canonical lexicon using well-established approaches from the LTAG literature (in particular (Xia et al., 2000): we traverse the converted tree from each leaf up towards the root, as long as the parental node is the head child of its parent. If a subtree is not the head child of its parent, we extract it as an elementary tree and proceed in this way for each word of the converted tree. Given the argument/modifier distinction, we then create substitution nodes in the parent tree for arguments or a root and foot node in the child tree for modifiers. Prediction trees are extracted automatically by calculating the minimal amount of structure needed to connect each word into a structure including all previous words of the sentence[2]. The parts of this
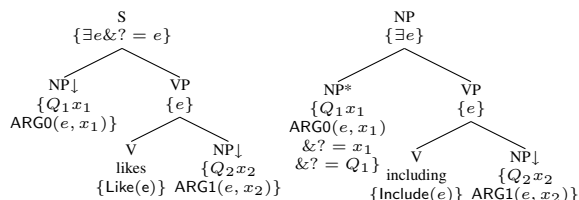
Figure 1: Verbal elementary trees extracted from example sentence *Pete likes sugary drinks including alcoholic ones.*

minimally-needed connecting syntactic structure which belong to heads to the right of the current word are stored in the lexicon as prediction trees, c.f. right hand side of figure 5b.

Since Propbank is used in the construction process of the PLTAG treebank, we can straightforwardly display the semantic role annotation on the tree and the extracted lexicon, with the exception that we display role annotations for PPs on their NP child. For arguments, annotations are retained on the substitution node in the parental tree, while for modifiers, the role annotation is displayed on the foot node of the auxiliary tree, as shown for the verbal trees extracted from the sentence *Pete likes sugary drinks including alcoholic ones* in Figure 1. PropBank assigns two roles to the NP node above *sugary drinks* (it is the `ARG1` of *likes* and the `ARG0` of *including*), but we can correctly tease apart these annotations in the lexical extraction process using the syntactic annotation and argument/modifier distinction.

Using the same procedure, prediction trees are annotated with semantic roles. It can then happen that one form of a prediction tree is annotated with different syntactic roles, hence introducing some additional ambiguity into the lexicon. For example, the NP substitution node in subject position of the prediction tree rooted in S_k in figure 5b could be an `ARG0` for some verbs which can verify this tree and an `ARG1` for others.

PLTAG elementary trees can contain one or more lexemes, where the first lexeme is the elementary tree's main anchor, and all further lexemes are predicted. In earlier PLTAG extractions, elementary trees with several lexemes were used for particle verbs like *show up* and some hand-coded constructions in which the first part is predictive of the second part, such as *either . . . or* or *both . . . and*. Here we extend this set of trees with

---

[1]Consider the optionality of the agent role in passive sentences, where the "by-phrase" may or may not appear.

[2]The reader is referred to (Demberg-Winterfors, 2010;

Demberg et al., 2014) for full details of the PLTAG conversion and syntactic part of the lexicon extraction process.

more than one lexeme to verbs with subcategorized PPs, as shown, for example, in the second lexicon entry of *slid* in figure 5a. Note the difference to the lexicon entry of optional PPs in figure 5b as in *on Sunday*. Furthermore,

- All elementary trees which have a role annotation in PropBank also have a corresponding annotation $\exists e$ on their root node that represents the existentially-quantified neo-Davidsonian event variable for that predicate, see fig. 1.

- The event variables and entity variables on an elementary tree are available for binding on the path from the anchor[3] of the elementary tree to the root node.

- Every role annotation on a node is in the form of a predicate $\text{ARG}n(e, x)$, where e is the event variable, and x is an entity variable to which the role is conferred.

- Every role annotation is prefixed with a variable binding Qx, where Q is a higher-order variable that represents an unknown quantifier. This ensures that all variables are bound if a role appears before its filler.

- Every elementary tree for an open-class word has a head with corresponding predicate. For example, "butter" has a predicate $\text{Butter}(x)$.

- Prediction trees for open lexical classes (such as NPs) have a head with a $\_\_(x)$ predicate.

- Every nominal elementary tree has a Qx at the root node so that the entity variable that is the argument to the predicate on the head is bound. The Qx is on the root node so that our semantic processing procedure for substitutions and adjunctions (described in the next section) can unify the entity variable x with variables on higher trees.

For PPs, we obtain role annotations from PropBank and NomBank. Other closed-class syntactic types such as pronouns have appropriately-selected quantifier constants and predicates (e.g. "someone" would be represented as $\exists x \text{Person}(x) \& ? = \exists \& ? = x$, see next paragraph for the use of question marks). Determiners are merely annotated with a quantifier "constant" symbol and no variables or predicates.

Then we require a type of additional annotation to which we refer as a "variable assignment statement", which we use in our syntactic com-

bination process. These statements are written $? = v$, where v is either a quantifier variable (Q) or constant (e.g. $\exists$) or an entity variable (x). These statements represent the possibility that an incoming tree might have a variable v that could have the same binding as one already in the prefix tree. Variable assignment statements occur on root nodes or foot nodes, except where there is a descendent DT subsitution node, which receives an additional $? = Q$ statement. The type of variable assignment statement (event, entity or quantifier) depends on the root node type (entity type like NP or N vs. event type like S or VP), as shown in figure 1. The next section describes the use of these statements in semantic parsing. Note that variable assignment statements need not be represented explicitly in an implementation, as reassigning variables can be done via references or other data structures. We use them as a representational and illustrative convenience here.

6.2 Semantic parsing procedure

We integrate semantics into the overall process of PLTAG parsing by the rules in figures 2 and 3. In addition, we provide a more procedural description here. At the highest level, a step in an incremental parse follows this pattern:

1. On scanning a new word or doing a prediction step, the PLTAG statistical model selects a tree from the lexicon, an operation (substition, adjunction, verification), and a position in the prefix tree at which to insert the tree (or none, if this is the first word).

2. All the nodes of the incoming tree are visited by the visit operation, and their semantic content is appended as conjuncts to the output semantic expression.

3. The operation of attaching the new tree into the derived tree is performed (pltagOp):

   (a) Variable assignment statements are emitted and appended to the semantic output expression according to the rules in figure 3, as well as to the semantic expression at the syntactic node at which the integration occurs. For verification, the Verify rule has to be applied to all nodes that are verified.

   (b) The syntactic integration of merging the nodes at the substitution or adjunction site is performed. The rules in 3 also make sure that the semantic expressions

---

[3]Lowest node on the path to where the anchor would be in a prediction tree which does not have a lexical anchor.

$$\frac{D : \{\Psi\} \qquad T}{\text{pltagOp}(D, T) : \{\Psi \& \text{visit}(T)\}} \text{ PltagStep} \qquad\qquad \frac{D : \{\Psi\}}{D : \text{resolveEqns}(\Psi)} \text{ Resolve}$$

Figure 2: Overall rules for trees (T) and derivations (D) and overall semantic expressions (Ψ). PltagStep applies when a new tree is chosen to be integrated with the prefix tree.

$$\frac{N_1 \Downarrow: \{\Sigma_1, Q_1, \Sigma_2\} \qquad N_2 \Uparrow: \{\Sigma_3, ? = Q_2, \Sigma_4\} \qquad D : \{\Psi\}}{D[N_1 \mapsto \text{nodeMerge}(N_1 : \{\Sigma_1, Q_1, \Sigma_2\}, N_2 : \{\Sigma_3, \Sigma_4\})] : \{\Psi \& Q_1 = Q_2\}} \text{ QuantEquate}$$

$$\frac{N_1 \Downarrow: \{\Sigma_1, x_1, \Sigma_2\} \qquad N_2 \Uparrow: \{\Sigma_3, ? = x_2, \Sigma_4\} \qquad D : \{\Psi\}}{D[N_1 \mapsto \text{nodeMerge}(N_1 : \{\Sigma_1, x_1, \Sigma_2\}, N_2 : \{\Sigma_3, \Sigma_4\})] : \{\Psi \& x_1 = x_2\}} \text{ VarEquate}$$

$$\frac{N_1{}_p^p : \{\Sigma_1, \underline{\ }_1, \Sigma_2\} \qquad N_2 : \{\Sigma_3\} \qquad \text{anchor}(N_2):\{\Sigma_4, Pred(x), \Sigma_5\} \qquad D : \{\Psi\}}{D[N_1 \mapsto \text{nodeMerge}(N_1 : \{\Sigma_1, \underline{\ }_1, \Sigma_2\}, N_2 : \{\Sigma_3\})] : \{\Psi \& \underline{\ }_1 = \text{Pred}\}} \text{ Verify}$$

Figure 3: Rules for combining nodes. The nodes are attached during the derivation via the nodeMerge operation, with N_1 being the node above ($\Downarrow$), and N_2 being the node below ($\Uparrow$). These hold for substitution and adjunction (for both canonical and prediction trees). The underlying intuition is that the ($\Downarrow$) node will contain the variable equation, and the ($\Uparrow$) node will contain the mention of a variable to be equated. The Verify rule equates the $\underline{\ }$ variable with the predicate of the verification tree. The equation is appended to the output expression Ψ. Q_2 can also be $\exists$ or another quantifier. VarEquate also applies to event variables. The Σ_n notation represents the prefixes and suffixes of the semantic expressions relative to the mentioned variable or statement. The rules delete the variable assignment statement from the node by concatenating Σ_3 and Σ_4.

from both nodes involved in the integration are included in the semantic expression of the merged node.

4. Optionally, a Resolve step is applied, which eliminates variable assignment statements by replacing variable mentions with their most concrete realization.

Regarding variable assignments at the integration of two trees, the value for quantifier variables can be a constant in the form of a quantifier. Entity variables can be equated with other entity variables, and entity constants (e.g., proper names) are a relatively simple extension to the rules[4]. Verification variables can only be equated with a constant—a predicate name.

We present an example of the processing of a substitution step in figure 4. The S tree for *sleeps* with an open NP substitution node is in the process of having the NP "someone" substituted into it using the substUp operation. So we have already done step 1 of our parsing procedure. Step 2 is visit, such that the semantic expression of the NP is appended to the output expression

Ψ. For step 3, the variable assignment statements are then processed by application of QuantEquate and VarEquate. Finally in step 4, the expression is simplified with Resolve.

The Resolve operation. From an implementation perspective, resolving variable assignment statements does not really need a separate operation, as references can be maintained such that the assignment is automatically performed without any explicit substitution in the manner of a Prolog inference engine's resolution procedure. The same holds for the variable assignment statements. However, we include explicit mention of this mechanism for ease of expression of the semantic operations as well as to illustrate some degree of convergence with existing formalisms such as (R)MRS, which also has a mechanism to assert relationships between variables *post hoc*.

There is only one condition under which application of Resolve can fail, which is if there is more than one assignment statement connecting the same variable to different constants.

The Resolve rule is defined to be able to apply to the entire output expression. When should it apply? It is defined such that it can be applied at any time; its actual execution will be controlled by the parsing algorithm, e.g., after each parsing operation or at the end of the parse.

There are remaining matters of quantifier scope

---

[4]A noun phrase like "Peter" will have the associated semantic expression peter&? = peter and will require an additional inference rule to remove the quantifier when it is adjoined or substituted to a node carrying a role. In other words, substituting peter into $Qx\text{ARG1}(e, x)$ should result in $\text{ARG1}(e, \text{peter})$. An analogous rule for constant verification that allows $Qx\underline{\ }(x)$ to be verified as peter is also required.
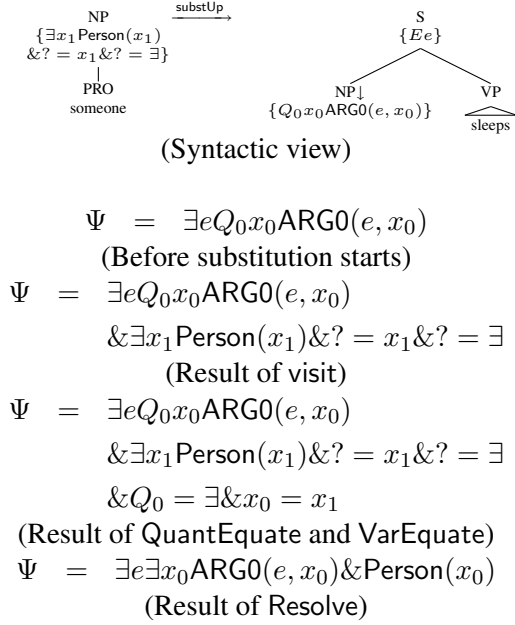
(Syntactic view)

$$\Psi = \exists e Q_0 x_0 \mathsf{ARG0}(e, x_0)$$

(Before substitution starts)

$$\Psi = \exists e Q_0 x_0 \mathsf{ARG0}(e, x_0)$$
$$\&\exists x_1 \mathsf{Person}(x_1)\&? = x_1\&? = \exists$$

(Result of visit)

$$\Psi = \exists e Q_0 x_0 \mathsf{ARG0}(e, x_0)$$
$$\&\exists x_1 \mathsf{Person}(x_1)\&? = x_1\&? = \exists$$
$$\& Q_0 = \exists \& x_0 = x_1$$

(Result of QuantEquate and VarEquate)

$$\Psi = \exists e \exists x_0 \mathsf{ARG0}(e, x_0)\&\mathsf{Person}(x_0)$$

(Result of Resolve)

Figure 4: An example incremental step from the semantic perspective.

and semantic well-formedness that must be handled *post hoc* at every step. For example, universal quantifiers require a distinction to be made between the restrictor of the quantified variable and the nuclear scope. It is possible within a neo-Davidsonian representation to perform such representational adjustments easily, as shown by Sayeed and Demberg (2012).

Example Now that we have described the procedure, we provide an example of how this semantic augmentation of PLTAG can represent role labeling and prediction inside the syntactic parsing system. We perform a relevant segment of the parse of example (1-a), "The woman slid the butter to the man." In this sentence, we expect that the parser will already know the expected role of the NP "the man" before it actually receives it. That is, it will know in advance that there is an upcoming NP to be predicted such that it is compatible with a recipient (ARG2) role, and this knowledge will be represented in the incremental output expression.

The minimum lexicon required for our example is contained in figures 5a and 5b. For our illustration, we only include the ditransitive alternation of "slide". Both versions of slide contain all the roles on NP nodes. This parse involves only the prediction of noun phrases, so we only have an NP prediction tree. We presume for the sake of simplicity that the determiner "the" represents the existential quantifier $\exists$.

Our parse begins in figure 5c with "The woman slid", since these are the same in both cases, and it proceeds up to figure 5e with the sentence "The woman slid the butter to the man". We Resolve the assignments at every step for brevity in the examples, and we also apply it to the nodes. By figure 5d, the parser already knows that the ARG2 of "slide" is what is sought. Finally, by figure 5e, the appropriate NP is expected by prediction.
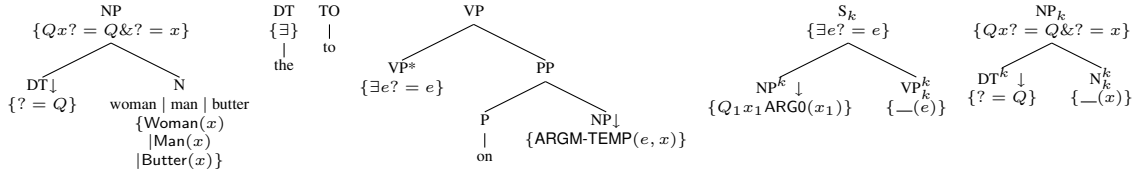
7 Discussion and conclusions

We demonstrated how syntactic prediction and thematic roles can interact in our framework, but we did so with a simple example of prediction: a single noun phrase. Our framework is, however, able to accomodate more complex interactions. In particular, we want to draw attention to an example which can not be modelled by other formalisms which are not fully connected like PLTAG. Consider sentences beginning with "The victim/criminal was violently...". Does the semantic association between "victim" vs. "criminal" and "violently" change the likelihoods of the semantic roles that can be assigned to the subject NP? Does it make an active or a passive voice verb more likely after "violently"? These are the kinds of possible syntactic-semantic interactions for which one will need a flexible but robust formalism such as we have described in this paper: the prediction mechanism allows dependents to jointly affect the expectation of a head even before the head has been encountered. Note that these interactions can also go beyond thematic roles.
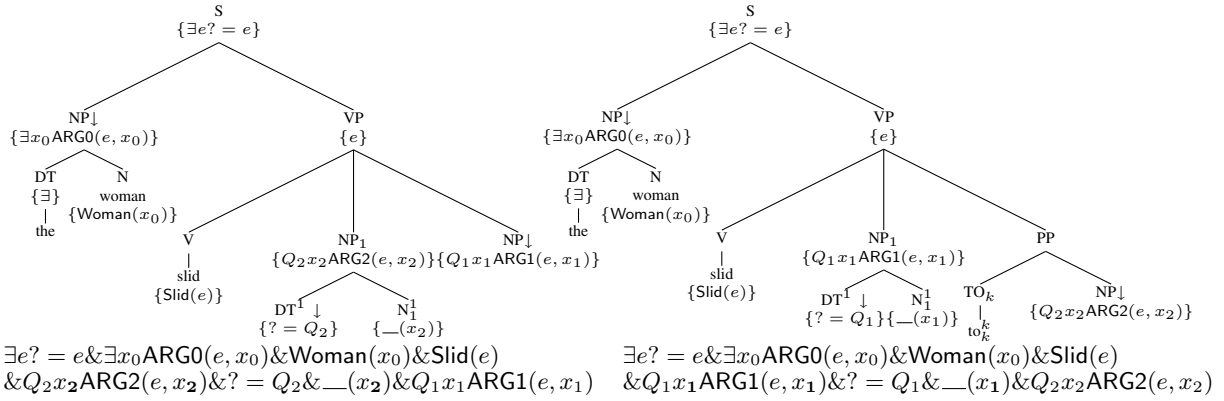
In this paper, we have presented a procedure to augment a treebank-extracted PLTAG lexicon with semantic annotations based in a flexible neo-Davidsonian theory of events. Then we have provided the way to combine these representations during incremental parsing in a manner fully synchronized with the existing PLTAG syntactic operations. We demonstrated that we can represent thematic role prediction in a case that is known to be relevant to an on-going stream of psycholinguistic research. Ongoing and future work includes the development of a joint syntactic-semantic statistical model for PLTAG and experimental validation of predictions made by our semantic augmentation. We are also considering higher-order semantic issues such as quantifier scope underspecification in the context of our formalism (Koller et al., 2003).
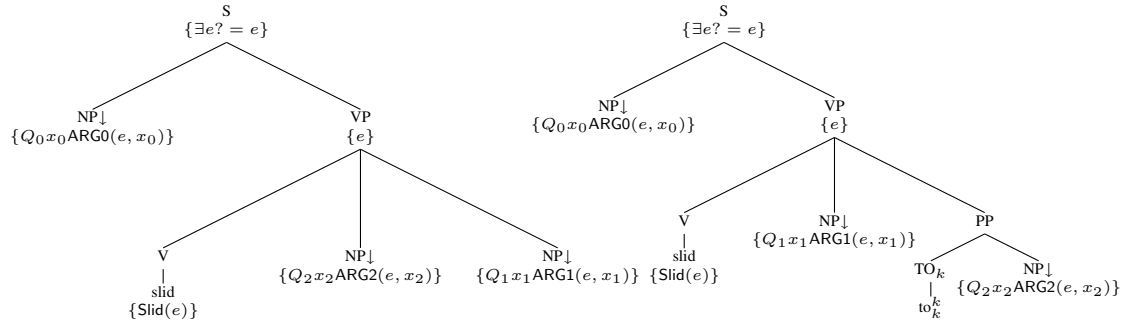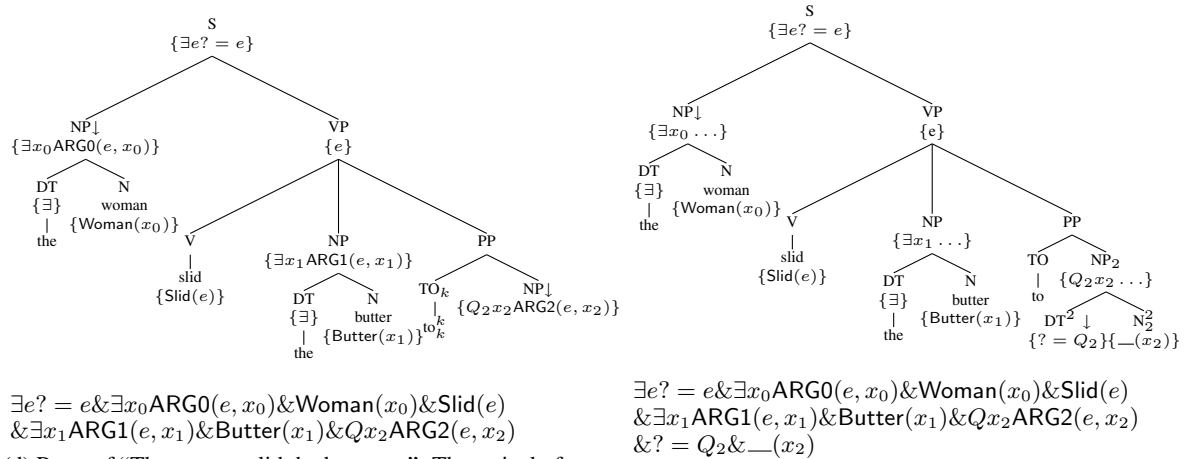
(a) Lexicon: ditransitive alternation of *slid*.

(b) Lexicalized trees and prediction trees.

$$\exists e? = e \& \exists x_0 \mathsf{ARG0}(e, x_0) \& \mathsf{Woman}(x_0) \& \mathsf{Slid}(e)$$
$$\& Q_2 x_2 \mathsf{ARG2}(e, x_2) \& ? = Q_2 \& \_\!\_(x_2) \& Q_1 x_1 \mathsf{ARG1}(e, x_1)$$

$$\exists e? = e \& \exists x_0 \mathsf{ARG0}(e, x_0) \& \mathsf{Woman}(x_0) \& \mathsf{Slid}(e)$$
$$\& Q_1 x_1 \mathsf{ARG1}(e, x_1) \& ? = Q_1 \& \_\!\_(x_1) \& Q_2 x_2 \mathsf{ARG2}(e, x_2)$$

(c) Parse of "The woman slid" with respect to the ditransitive alternation, with the syntactic prediction of an NP. Two possibilities still remain. The semantics are identical except for the role of the predicted nominal predicate. The $? = e$ variable assignment statement persists through the derivation, representing the possibility that this sentence is embedded in another.

$$\exists e? = e \& \exists x_0 \mathsf{ARG0}(e, x_0) \& \mathsf{Woman}(x_0) \& \mathsf{Slid}(e)$$
$$\& \exists x_1 \mathsf{ARG1}(e, x_1) \& \mathsf{Butter}(x_1) \& Q x_2 \mathsf{ARG2}(e, x_2)$$

(d) Parse of "The woman slid the butter...". The arrival of "the butter" greatly reduces the likelihood of the recipient role (ARG2) being the one filled at this point, effectively abolishing the first parse.

$$\exists e? = e \& \exists x_0 \mathsf{ARG0}(e, x_0) \& \mathsf{Woman}(x_0) \& \mathsf{Slid}(e)$$
$$\& \exists x_1 \mathsf{ARG1}(e, x_1) \& \mathsf{Butter}(x_1) \& Q x_2 \mathsf{ARG2}(e, x_2)$$
$$\& ? = Q_2 \& \_\!\_(x_2)$$

(e) Parse of "The woman slid the butter to...". *to* is verified and the last NP is expanded via prediction. This gives us the last predicted predicate in the semantic expression. It shares its variable with the ARG2 role, thus thematically restricting its possible verifications.

Figure 5: Excerpt of our example parse.

References

Gerry Altmann and Yuki Kamide. 1999. Incremental interpretation at verbs: Restricting the domain of subsequent reference. *Cognition*, 73(3):247–264.

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

Vera Demberg and Frank Keller. 2008. A psycholinguistically motivated version of tag. In *Proceedings of the 9th International Workshop on Tree Adjoining Grammars and Related Formalisms. Tübingen*, pages 25–32.

Vera Demberg and Frank Keller. 2009. A computational model of prediction in human parsing: Unifying locality and surprisal effects. In *Proceedings of the 29th meeting of the Cognitive Science Society (CogSci-09)*.

Vera Demberg, Frank Keller, and Alexander Koller. 2014. Parsing with psycholinguistically motivated tree-adjoining grammar. *Computational Linguistics*, 40(1).

Vera Demberg-Winterfors. 2010. *A Broad-Coverage Model of Prediction in Human Sentence Processing*. Ph.D. thesis, University of Edinburgh.

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

Srini Narayanan Daniel Jurafsky. 2002. A bayesian model predicts human parse preference and reading times in sentence processing. In *Advances in Neural Information Processing Systems 14: Proceedings of the 2001 Neural Information Processing Systems (NIPS) Conference*, volume 1, page 59. The MIT Press.

Yuki Kamide, Gerry Altmann, and Sarah L Haywood. 2003a. The time-course of prediction in incremental sentence processing: Evidence from anticipatory eye movements. *Journal of Memory and Language*, 49(1):133–156.

Yuki Kamide, Christoph Scheepers, and Gerry TM Altmann. 2003b. Integration of syntactic and semantic information in predictive processing: Cross-linguistic evidence from german and english. *Journal of Psycholinguistic Research*, 32(1):37–55.

Alexander Koller, Joachim Niehren, and Stefan Thater. 2003. Bridging the gap between underspecification formalisms: Hole semantics as dominance constraints. In *Proc. of EACL 2003*, pages 367–374.

David M Magerman. 1994. *Natural language parsing as statistical pattern recognition*. Ph.D. thesis, Stanford University.

Jeff Mitchell, Mirella Lapata, Vera Demberg, and Frank Keller. 2010. Syntactic and semantic factors in processing difficulty: An integrated measure. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 196–206. Association for Computational Linguistics.

Ulrike Padó, Matthew W Crocker, and Frank Keller. 2009. A probabilistic model of semantic plausibility in sentence processing. *Cognitive Science*, 33(5):794–838.

Martha Palmer, Dan Gildea, and Paul Kingsbury. 2003. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

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

Philip Resnik. 1992. Left-corner parsing and psychological plausibility. In *In The Proceedings of the fifteenth International Conference on Computational Linguistics, COLING-92*, pages 191–197.

Brian Roark. 2001. Probabilistic top-down parsing and language modeling. *Computational linguistics*, 27(2):249–276.

Asad Sayeed and Vera Demberg. 2012. Incremental neo-davidsonian semantic construction for tag. In *11th Workshop on Tree-Adjoining Grammars and Related Formalisms (TAG+11)*.

Mark Steedman. 2000. *The syntactic process*. MIT Press.

Henry S. Thompson, Mike Dixon, and John Lamping. 1991. Compose-reduce parsing. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pages 87–97, Berkeley, California.

David Vadas and James Curran. 2007. Adding noun phrase structure to the Penn Treebank. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 240–247, Prague, Czech Republic, June. Association for Computational Linguistics.

Fei Xia, Martha Palmer, and Aravind Joshi. 2000. A uniform method of grammar extraction and its applications. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 53–62.