A Psycholinguistically Motivated Version of TAG

Vera Demberg and **Frank Keller**
School of Informatics, University of Edinburgh
2 Buccleuch Place, Edinburgh EH8 9LW, UK
{v.demberg,frank.keller}@ed.ac.uk

Abstract

We propose a psycholinguistically motivated version of TAG which is designed to model key properties of human sentence processing, viz., incrementality, connectedness, and prediction. We use findings from human experiments to motivate an incremental grammar formalism that makes it possible to build fully connected structures on a word-by-word basis. A key idea of the approach is to explicitly model the prediction of upcoming material and the subsequent verification and integration processes. We also propose a linking theory that links the predictions of our formalism to experimental data such as reading times, and illustrate how it can capture psycholinguistic results on the processing of *either* ... *or* structures and relative clauses.

1 Introduction

Current evidence from psycholinguistic research suggests that language comprehension is largely *incremental*, i.e., that comprehenders build an interpretation of a sentence on a word-by-word basis. This is a fact that any cognitively motivated model of language understanding should capture. There is also evidence for full *connectivity* (Sturt and Lombardo, 2005), i.e., for the assumption that all words are connected by a single syntactic structure at any point in the incremental processing of a sentence. While this second point of full connectivity is more controversial, the model we are proposing here explores the implications of incrementality in its strict interpretation as full connectivity.

Furthermore, recent work on human sentence comprehension indicates that people make *predictions* of upcoming words and structures as they process language (Frazier et al., 2000; Kamide et al., 2003; Staub and Clifton, 2006). The concepts of connectedness and prediction are closely related: in order to assure that the syntactic structure of a sentence prefix is connected at every point in time, it can be necessary to include phrases whose yield has not been processed yet. This part of the structure needs to be generated by the parser in order to connect the words that have been seen so far, i.e., to achieve full connectivity (which in turn is required to build an incremental interpretation). This process has been formalized by (Lombardo and Sturt, 2002) using the notion of *connection path*.

In this paper, we explore how these key psycholinguistic concepts (incrementality, connectedness, and prediction) can be realized within a new version of tree-adjoining grammar (TAG), which we call Psycholinguistically Motivated TAG (PLTAG). We argue that TAG is better suited for this modeling task than other formalisms such as CCG or PCFGs and propose a linking theory that derives predictions of processing difficulty from aspects of the PLTAG formalism.

2 Related Work

A number of incremental versions TAG have been proposed over the years (Shen and Joshi, 2005; Kato et al., 2004; Mazzei et al., 2007). The version proposed here differs from these approaches in a number of ways. Spinal LTAG (Shen and Joshi, 2005) does not implement full connectivity, and cannot easily be used to model prediction since it does not encode valencies. The proposals by (Mazzei et al., 2007) and (Kato et al., 2004) are more similar to our work, but are less well-suited for psycholinguistic modeling since they do not implement a verification mechanism, which is required to account for standard complexity results in the spirit of (Gibson, 1998). In addition, (Kato et al., 2004) do not distinguish between modifiers and arguments, since they operate on the Penn Treebank, where this information is not directly available.

Incremental parsers for other grammar formalisms include Roark's (2001) for PCFGs, and Nivre's (2004) for dependency grammars. Neither of these parsers implement strict incrementality, in the sense of always building connected structures. Furthermore, there are principled problems with PCFGs are a model of prediction difficulty, even if fully connected structures are built (see Section 7).

The main contributions in this version of TAG introduced in this paper is that it is incremental and respects full connectivity, whiles also modeling the verification and integration of syntactic material. Our main emphasis is on the modeling of prediction, which has been the subject of much recent research in psycholinguistics, as outlined in the previous section.

3 Incrementality and Prediction

We propose variant of TAG that incorporates two different types of prediction: prediction through substitution nodes in lexicon entries (e.g., if a verb subcategorizes for an object which has not yet been seen), and prediction via connection paths. The first type of prediction models the anticipation of upcoming syntactic structure that is licensed by the current input; the second type models prediction which is required to ensure that fully connected structures are built. We will discuss the mechanism for prediction due to connectivity first.

3.1 Prediction due to Connectivity

TAG elementary trees can not always be connected directly to a previously built syntactic structure. Examples are situations where two dependents precede a head, or where a grandparent and a child have been encountered, but the head of the parent node has not. For instance, in the sentence *the horse seldom fell*, the elementary tree of *the horse* cannot directly be combined with elementary tree of the adverbial modifier *seldom*, see Figure 1(a). The head *fell* which provides the intervening structure, has not been encountered at that point. Therefore, this intervening structure has to be predicted in order to connect *the horse* and *seldom*.[1] We use the substitution symbol $\downarrow$ to mark predicted structure. As a prediction mark, the substitution symbol can therefore also occur tree-internally. We assume that prediction is conservative, and only includes the structure as far as it is needed, i.e., only as far as it is included in the connection path (see Section 4 and Figure 3). It is important to bear in mind, however, that prediction grain size remains an open research question (for instance, we could predict the full elementary tree down to the lexical item, as proposed by (Mazzei et al., 2007), to even include the remaining subcategorized nodes or likely modifiers of that node).

Our minimal prediction method implies that adjunction must be possible at predicted nodes, as shown in Figure 1(a). When this happens, the head node of the auxiliary tree is marked as seen, while the foot node of the auxiliary tree takes over the prediction mark from the predicted connection structure, because we need to mark that we have not in fact yet seen the node that it adjoined to. If we marked both as non-predicted nodes, then we would not be able to guarantee that we can correctly keep track of what has been encountered in the input and what we have predicted.

We treat those connecting structures as special lexicon entries, where each predicted node is marked. A predicted node differs from the rest of the structure in that it needs to be verified, i.e., it has to be matched (through substitution of internal nodes) with later upcoming structure, as illustrated in Figure 1(b). A derivation of a sentence is only valid if all predicted nodes are matched. Our example shows how the tree structure for *the horse seldom* is connected with the elementary tree of *fell*. Each node of the new elementary tree can either be matched with a predicted node in the prefix tree, or it can be added (the structure for *the horse seldom*. It could therefore just as easily unify with a transitive or ditransitive verb. (Note that by unification we simply mean node matching and we will use these two terms interchangeably in this paper.)

Issues arise in the verification process, e.g., how to unify structures after additional material has been adjoined. In our example, an additional VP node has been introduced by the adverb. The new nodes in the tree cannot unify with a random predicted node of the same category, but have to follow constraints of accessibility and have to have identical dominance relations. For example, consider a situation where we predict the structure between an object relative pronoun like *whom* and its trace (see the top tree in Figure 4). If we encountered a verb next, we could match up the nodes of the verb el-

[1] Because of the recursiveness of natural language, it is possible that there are infinitely many ways to connect two trees. Although embedding depth can be infinite in theory, we assume that it is finite and indeed very small due to limitations of human memory.

ementary tree (S, VP, V) with the predicted nodes, and would still predict the subject noun phrase. If we then encountered a noun phrase, and again did not take into account any accessibility constraints (the substitution node is not accessible any more because filling it at this point would violate the linear order), we could substitute that noun into the subject position. That is, we would accept impossible RCs like *whom thanked Peter*, or misanalyze subject relative clauses as object relative clauses.

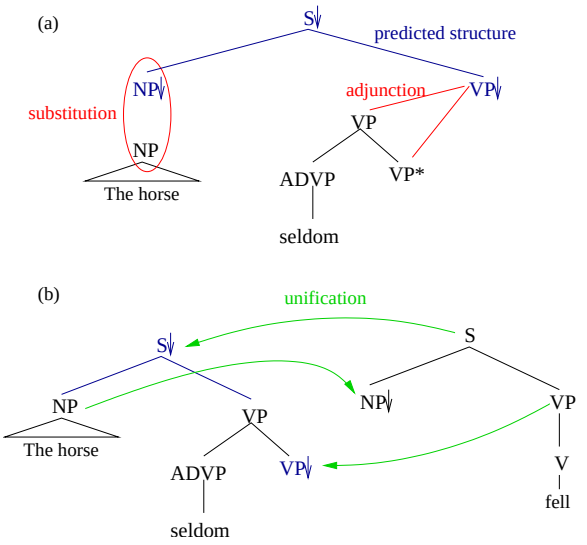


Figure 1: Example for prediction and verification of predictions

3.2 Prediction from Substitution Nodes

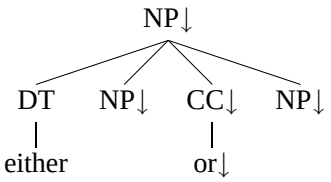
Another source for predictions are the lexicon entries themselves. Each substitution node that is to the right of the tree's anchor naturally becomes a prediction during the parsing process. This means that we do not predict modifiers or any other kind of recursive structures, unless we have already seen a word that depends on the modifier (i.e., through connectivity, e.g., for a sentence prefix such as *the horse very*). Whether or not modifiers are predicted syntactically is currently an open research question. Preliminary evidence suggests that modifiers are predicted when they are required by the discourse context.

We also exploit TAG's extended domain of locality in order to construct lexicon entries such that are more appropriate for modeling psycholinguistic findings. An example is the *either* ... *or* construction. Results by (Staub and Clifton, 2006) show that hearing the word *either* triggers prediction of *or* and the second conjunct: reading times on these regions were shorter in the *either* condition, and participants also did not misanalyze disjunctions at sentence level as noun disjunctions in the condition where *either* was present.

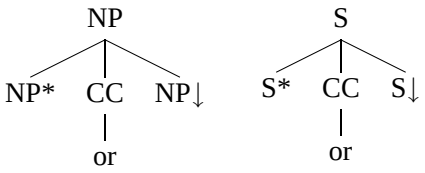
As (Cristea and Webber, 1997) point out, there are a number of constructions with two parts where the first part can trigger prediction of the second part in, similar to *either* ... *or*. A related form of prediction is syntactic parallelism; experimental findings by (Frazier et al., 2000) indicate that the second conjunct of a coordinate structure is processed faster if its internal structure is identical to that of the first conjunct. This can be seen as a form of prediction, i.e., the parser predicts the structure of the second conjunct as soon as it has processed the conjunction.

Here, we will discuss in more detail how *either* ... *or* prediction can be implemented our framework. We assign a lexicon entry to *either* which predicts the occurrence of coordination with two entities of the same category, and requires *or* as a coordinator, see Figure 2(a). Figure 2 shows an example of how the word *either* impacts parsing of the *either* ... *or* disjunction in PLTAG, as opposed to a simple *or* disjunction. In the no *either* case, a sentence structure like Figure 2(c) can be combined with either of the elementary tree of *or*, as shown in Figure 2(b). Two different analyzes are created, one for the noun disjunction case and one for the sentence disjunction. Later on in processing, one of these is ruled out when disambiguating information is processed. The position of *either* can help disambiguate this ambiguity before it arises, which explains why participants were not misanalyzing sentence disjunctions when *either* was present. Furthermore, changes in the probabilities of the analyses occur at different points in time in the *either* and no *either* cases. Structures that have been predicted and do not add any new nodes incur integration costs but do not cause any changes in probabilities of the analysis.

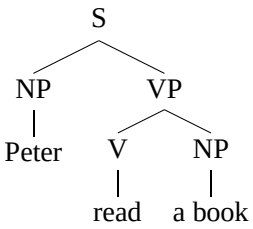
In this case, *either* and *or* provide overlapping information, in particular, *or* does not give any new information. This means that we either have to have a different lexicon entry for *or* following *either*, or that adjunction works differently for those partly redundant nodes. Because both the foot and the head node of *or* have been predicted by previously by *either*, the head node of the auxiliary tree just verifies the prediction, while the foot node adopts whatever annotation is on the node



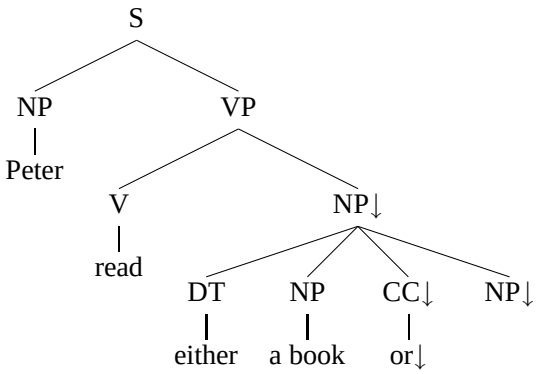
(a) lexicon entry for *either*



(b) lexicon entries for *or*



(c) no *either*



(d) with *either*

Figure 2: Example for the use of TAG's extended domain of locality to model expressions that trigger predictions, such as *either . . . or*

that it matches. This situation can be automatically recognized because the lexical anchor for the *or*-auxiliary tree was itself predicted. Also note that in this case, the missing second conjunct gets marked twice for substitution (both by the lexicon entry for *either* in Figure 2(a) and 2(b)). This double prediction changes the time stamp on the predicted node, which gets set to the most recent time it was predicted.

This kind of redundancy by eager prediction also occurs in our analysis of relative clauses. In theory, encountering the object relative pronoun *whom* is sufficient to predict the argument structure of the relative clause (namely that there has to be a head for the relative clause, and that there has to be a subject, and a trace for the object). We will investigate in future work whether there is evidence that humans predict the whole structure given the relative pronoun. For now we assume that a trace is always predicted when its filler is encountered. For an example of how this works, see Figure 4.

4 Treebank-based Lexicon Induction

We induce the lexicon needed for our incremental version of TAG from the Penn Treebank, complemented by Nombank (Vadas and Curran, 2007) and Propbank (Palmer et al., 2003), as well as Magerman's head percolation table (Magerman, 1994). These additional resources help determine the elementary trees and distinguish arguments from modifiers. (Modifiers are not predicted unless they are needed for a connection path.) In Figure 3 each inner node is indexed with the number of the word that is its lexical anchor in order to show which parts of the syntactic tree belong to which lexicon entry.

Once the parsed trees have been segmented into elementary trees (following procedures in Xia et al. 2000), we calculate connection paths for each prefix, as proposed by (Lombardo and Sturt, 2002). A connection path for words $w_1 \dots w_n$ is the minimal amount of structure that is needed to connect all words $w_1 \dots w_n$ into the same syntactic tree. The amount of structure needed at each word for the sentence *the Italian people often vote Berlusconi* is indicated in Figure 3 by the structure enclosed in the circles.

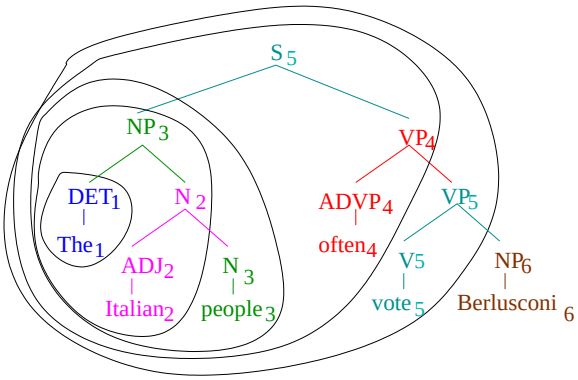


Figure 3: Generating lexicon entries from the Penn Treebank for an example sentence

We then use the connection paths and the canonical elementary trees to determine which parts of the structure that are included in the connection path for words $w_1 \dots w_n$, but not part of any of the elementary trees with feet $w_1 \dots w_n$. In Figure 3, this occurs twice: firstly when *Italian* has been read, and the determiner and adjective can only be combined by predicting that they must be part of the same noun phrase, and secondly at *often*, when the VP and S nodes have to be predicted.

By definition, all nodes of these connecting structures are predicted nodes, and therefore annotated as substitution nodes. We store these connecting structures as non-lexicalized lexicon entries. They differ from other lexicon entries in that all their nodes are substitution nodes, and in that they are not lexicalized. The advantage of generating these separate non-lexicalized entries over simply adding a second predicted version of all lexicon entries is that we retain a smaller lexicon, which reduces the sparse data problem for training, and makes parsing more efficient.

The connection structure is non-lexicalized, and therefore creates additional challenges for the parser: the non-lexicalized structures can be triggered at any point in parsing, in particular when simple substitution and adjunction are not successful. They can also in principle be chained, i.e., several of non-lexicalized structures can be applied one after the other, without ever applying any lexicalized rules. As a first approximation, we therefore restrict these prediction rules to instances that we encountered in the corpus, and do not only allow several non-lexicalized rules in a row. This restriction means that there may be sentences which this incremental parser cannot cover, even though a non-incremental parser (or one without this restriction) can find an analysis for them. (CCG has a similar problem with the application of type-raising; in current CCG parsers, the search problem in type-raising is solved by lexicalizing type raising.) Because of recursive rules in natural language, embedding can in principle also be infinitely deep. However, (Lombardo and Sturt, 2002) have shown that for 80% of the word tokens, no connection paths are needed, and that two or more predictions have to be made for about 2% of the tokens.

5 Linking Parsing Complexity to Processing Difficulty

The grammar design proposed here implements a specific set of assumptions about human language processing (strong incrementality with full connectedness, prediction, ranked parallel processing) which can be tested by linking an incremental parser for this formalism with a theory of human sentence comprehension.

The relation between the incremental parsing algorithm and processing difficulty can be formalized as follows: At each word, a set E of syntactic expectations e is generated (they can be easily read off the syntactic structure in the form of substitution nodes). These expectations can be interpreted as denoting the categories needed to build a grammatical sentence from the current input, and are associated with probabilities $P(e)$, estimated by the parser. Each structure also has a timestamp corresponding to when it was first predicted, or last activated. Based on this, decay is calculated, under the assumption that recently-accessed structures are easier to access and integrate (decay is weighted for verification (substitution of inner nodes), regular substitution and, adjunction).

In this model, processing difficulty is incurred either when expectations are incompatible with the current input (algorithmically, this corresponds to the parser trying to substitute, adjoin, or unify a new tree with the currently maintained structure, but failing for all structures), or when successful integration takes place (i.e., unification of predicted nodes and the elementary tree is successful, or a node can be successfully adjoined). Intuitively, integration is costly because the parser has to bring together the meaning of the matched categories.

Processing difficulty is proportional to the inverse probability of all integrated structures (less activated structures are harder to integrate) plus the probability of all deleted structures (more probable structures are harder to discard), where both probabilities weighted by recency:

$$D_w \propto \sum_{e \in E_i} f(\frac{1}{P(e)}) + \sum_{e \in E_d} f(P(e))$$

Here, D_w is the difficulty at word w, and E_i is the set of expectations that could be integrated, while E_d is the set of expectations that have been discarded at w. A decay is implemented by the function f.

6 Example

The following example aims to show how PLTAG can explain increased processing difficulty at object relative clauses (ORC) as opposed to subject relative clauses (SRC). We chose this example because there is evidence that object relative clauses are more difficult for humans to process from both experimental sources (King and Just, 1991; Gibson, 1998) and broad-coverage corpus data (Demberg and Keller, 2007).

Figure 4 shows two alternative structures for the phrase *grand-parents who* (all probabilities in this example are fictitious and just for illustrative purposes). The two analyses differ by whether they analyze *who* as an object relative pronoun or a subject relative pronoun, and predict traces in different positions. (Whether traces should be predicted when their fillers are encountered is an open question, but we will assume that they are for the time being.) Both of these analyses have a certain probability, which is higher for the SRC (0.0003) than for the ORC (0.00004), since SRCs are more frequent. When the next word is encountered, that word may also be ambiguous, such as the word *time* in our example, whose probability is higher as noun (0.08) than as a verb (0.02). All possible elementary trees for the new word have to be matched up with all prefix trees (analyses whose probability is below a certain threshold are ignored to limit the search problem and simulate memory limitations). In our example, the noun interpretation of *time* is compatible with the object relative clause interpretation, while the verb interpretation can be unified with the SRC analysis. The ORC structure still has lower probability than the SRC structure at this point, because $0.00003 \cdot 0.08 < 0.0004 \cdot 0.02$. If an ORC verb was encountered next, we would correctly predict that this verb should be more difficult to process than the SRC verb, because five nodes have to be matched up instead of four, and the predicted nodes in the ORC analysis are one clock-cycle older than the ones in the SRC at the time of integrating the verb.

On encountering a disambiguating word, the processing difficulty proportional to the probability mass of all incompatible structures would be incurred. This means that higher processing difficulty occurs when the more probable structure (the SRC in our example) has to be discarded.

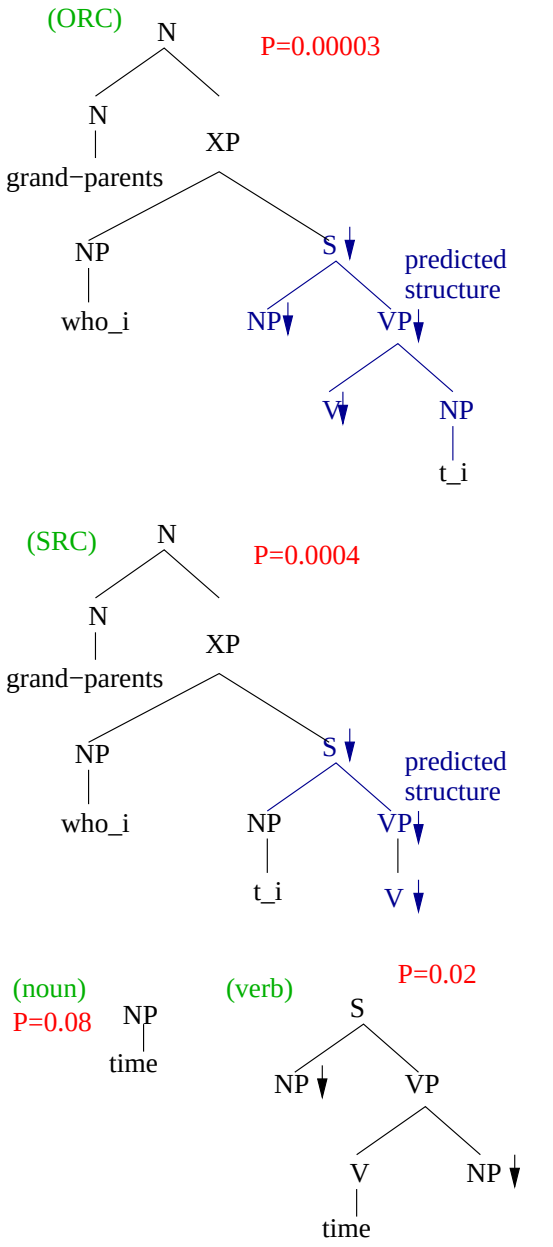


Figure 4: Example of the interaction of lexical probabilities and verification cost in PLTAG

7 Comparison with Other Grammar Formalisms

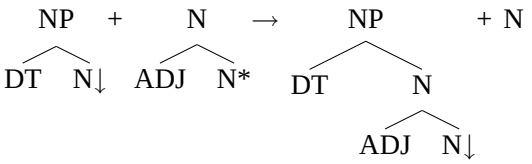
We decided to use tree-adjoining grammar instead of alternative formalisms like Combinatory Categorial Grammar (CCG) or Probabilistic Context Free Grammar (PCFG) because we felt that TAG best met our requirements of strict incrementality with full connectivity.

In standard CCG with bottom-up parsing (Steedman, 2000), it is not possible to always find an incremental derivation. For example, in ob-

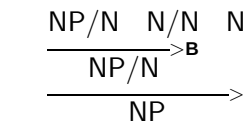
ject relative clauses, the subject NP of the relative clause cannot be integrated in an incremental fashion because the category of the relative pronoun ($(N\backslash N)/(S/NP)$) is too abstract: it does not contain the category for the subject NP explicitly and the subject NP therefore has to connect with the verb first. Another example are coordinated clauses. The second conjunct can only be combined with the first conjunct when they both have the same category. However, (Sturt and Lombardo, 2005) show that human sentence processing is more incremental than the most incremental CCG derivation for a sentence like *the pilot embarrassed John and put himself/herself in an awkward situation*, where the c-command relation between *the pilot* and *himself/herself* is understood at the point of reading the reflexive pronoun, and not only after reading the full second conjunct, as CCG would predict under the assumption that the syntactic relation has to be established first in order to determine c-command relations.

Coordination in tree-adjoining grammar does not have this problem of connecting with the beginning of the sentence only once the second conjunct has been seen, because the elementary tree for *and* is an auxiliary tree and adjoins into the previous structure, and therefore is connected to the preceding context right away, and *himself* can be substituted into the connected structure and is c-commanded by *the pilot* right away and will therefore be available for binding at an early processing stage.

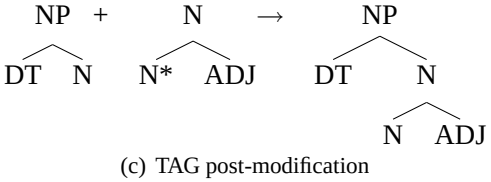
Furthermore, pre- and post-modification is asymmetric for incremental derivations in CCG (and we are not aware of such an asymmetry in human sentence processing). CCG requires either type-raising at a noun that comes before a modifier, or non-connectivity. The reason for the asymmetry is that for pre-modification, e.g., an adjective before noun, there is no type-raising necessary in incremental processing (see Figure 5(b)). On the other hand, for post-modification it is necessary to type-raise the head before the post-modifier is processed (see Figure 5(d)). This would lead to the unintuitive situation of having an ambiguity for a noun when it is post-modified, but not when it is pre-modified. Alternatively, the structure either has to be undone once the modifier is encountered in order to allow for the composition (serial account), or the noun is explicitly ambiguous as to whether it will be modified or not (parallel account), or we cannot satisfy full connectivity. In both cases, post-modification requires more operations than pre-modification. This is not the case in TAG, because pre- and post-modification are adjoined into the tree in the same fashion (see Figure 5(a) and (c)).



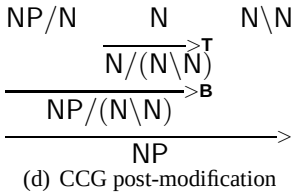
(a) TAG pre-modification



(b) CCG pre-modification



(c) TAG post-modification



(d) CCG post-modification

Figure 5: Comparision of pre- and post-modification in TAG and CCG

In order to use PCFGs as a basis for the psycholinguistic model it would be necessary to introduce composition into the parsing process in order to avoid having to predict all the processing difficulty at the end of phrases. Standard arc-eager parsing would for example complete a rule only once all of its children have been seen. For a more in-depth discussion of this question see (Thompson et al., 1991). Composition is also needed to keep track of the predictions. For example, once we have seen the verb, we do not want to expect the verb phrase itself anymore, but only any potential arguments. Furthermore, PCFGs do not provide the extended domain of locality that we exploit in TAG.

8 Summary

We propose a framework for a new version of TAG which supports incremental, fully connected derivations, and makes explicit predictions about upcoming material in the sentence. This version of TAG can be combined with a linking theory to model human processing difficulty, and aims to account for recent findings on prediction and connectivity in human sentence comprehension.

References

Cristea, Dan and Bonnie Webber. 1997. Expectations in incremental discourse processing. In *Proceedings of the 8th. Conference of the European Chapter of the Association for Computational Linguistics (ACL-EACL97)*, pages 88–95, Madrid, Spain, July. Association for Computational Linguistics.

Demberg, Vera and Frank Keller. 2007. Eye-tracking evidence for integration cost effects in corpus data. In *Proceedings of the 29th Annual Conference of the Cognitive Science Society*, Nashville.

Frazier, Lyn, Alan Munn, and Charles Clifton. 2000. Processing coordinate structure. *Journal of Psycholinguistic Research*, 29:343–368.

Gibson, Edward. 1998. Linguistic complexity: locality of syntactic dependencies. *Cognition 68*, pages 1–76.

Kamide, Yuki, Christoph Scheepers, and Gerry T.M. Altmann. 2003. Integration of syntactic and semantic information in predictive processing: Cross-linguistic evidence from German and English. *Psycholinguistic Research*, 32.

Kato, Yoshihide, Shigeki Matsubara, and Yasuyoshi Inagaki. 2004. Stochastically evaluating the validity of partial parse trees in incremental parsing. In Keller, Frank, Stephen Clark, Matthew Crocker, and Mark Steedman, editors, *Proceedings of the ACL Workshop Incremental Parsing: Bringing Engineering and Cognition Together*, pages 9–15, Barcelona, Spain, July. Association for Computational Linguistics.

King, J. and M. A. Just. 1991. Individual differences in syntactic processing: The role of working memory. *Journal of Memory and Language*, 30:580–602.

Lombardo, Vincenzo and Patrick Sturt. 2002. Incrementality and lexicalism. In *Lexical Representations in Sentence Processing, John Benjamins: Computational Psycholinguistics Series*, pages 137– 155. S. Stevenson and P. Merlo.

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

Mazzei, Alessandro, Vincenzo Lombardo, and Patrick Sturt. 2007. Dynamic tag and lexical dependencies. *Research on Language and Computation, Foundations of Natural Language Grammar*, pages 309–332.

Nivre, Joakim. 2004. Incrementality in deterministic dependency parsing. In *Proceedings of the ACL Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pages 50–57, Barcelona.

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

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

Shen, Libin and Aravind K. Joshi. 2005. Incremental ltag parsing. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 811–818.

Staub, Adrian and Charles Clifton. 2006. Syntactic prediction in language comprehension: Evidence from either...or. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 32:425–436.

Steedman, Mark. 2000. *The syntactic process*. The MIT press.

Sturt, Patrick and Vincenzo Lombardo. 2005. Processing coordinate structures: Incrementality and connectedness. *Cognitive Science*, 29:291–305.

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

Vadas, David 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.

Xia, Fei, Martha Palmer, and Aravind Joshi. 2000. A uniform method of grammar extraction and its applications. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora*, pages 53–62.