Tree-Grammar Linear Typing for unified Super-Tagging/Probabilistic Parsing Models

Ariane Halber
halber@enst.fr
AI group, dept of Computer-Science, ENST-Paris*
Man-Machine Interaction lab, Corporate Research Laboratory, Thomson-CSF†

Abstract

We integrate super-tagging, guided-parsing and probabilistic parsing in the framework of an item-based LTAG chart parser. Items are based on a linear-typing of trees that encodes their expanding path, starting from their anchor.

1 Introduction

Practical implementations of LTAG parsing have to face heavy lexical ambiguity and parsing combinatorial ambiguity. Main techniques to address these issues are **super-tagging** (Joshi and Srinivas, 1994), which consists in disambiguating elementary trees before parsing; **guided-parsing**, like head-driven parsing (van Noord, 1994) or anchor driven parsing (Lavelli and Satta, 1991; Lopez, 1998); and **probabilistic parsing** (Schabes, 1992; Caroll and Weir, 1997).

All of these approaches exploit specific properties of LTAG to improve parsing efficiency, but none is totally satisfactory.

Guided-parsing is a very nsefull means to limit overgeneration of spurious items in the chart, but it does not provide a new ambiguity bound. Besides, lexical ambiguity remains the main factor of computational load and this problem is only undirectly addressed by such techniques.

Super-tagging strength is to discard elementary trees while avoiding to go through actual tree combinations. It exploits instead local models of Well-Formedness (WF), as those used for tagging, where parse depencies remain implicit or underspecified. The problem though is that if a single tree is incorrect the parse will fail. To be robust, parsing must thus take several hypothesis into account. This leaves one with two regrets: first, parsing has still to find some way to tackle combinatorial ambiguity, and second, there is a lack of synergy between super-tagging and parsing , while they seem to share a kuowledge about tree potential-combinations.

Probabilistic parsing offers a way to tune the compromize between accuracy and speed, by thresholding partial parsing paths according to their current Inside probability. This incurs a well-known bias (Goodman, 1998): At a given derivation step, the Inside-probabilities of parse constituents estimate the relevance of the derivation *past*, but do not tell anything about its *future*. This can be corrected by A^* cost functions, or Outside-probability estimates.

To meet the weak points mentionned above, at least partialy, we develop a unified framework for the three techniques, and push their interactions further.

Sharing a parsing framework We propose an item-based chart-parser, where the parsing scheme is expressed as a deduction system (Shieber, Schabes, and Pereira, 1994). This framework is also amenable for expressing probabilistic parsing (Goodman, 1998).

Sharing models for super-tagging and item-pruning. Super-tagging can be seen as a tree-sequence WF-model, and extended to score derived item-sequences in the chart, wrt their likelihood of completing a parse. This yields a sound thresholding technique (Rayner and Carter, 1996; Goodman, 1998).

Sharing tree-types for item-pruning and guided-parsing. To support the WF parametric model, trees and items are abstracted by their linear type, which consists in a list of *connectors* that represent combination properties. Guided-parsing relies on a specific ordering of these connectors, so that a single type drives the parsing deduction and

*ENST Paris, 46 rue Barrault, 75634 Paris Cedex 13
†Thomson-CSF, LCR, Domaine de Corbeville, 91404 Orsay Cedex, FRANCE

Item form:	$<\Gamma_l \mid \Gamma_r>_{\alpha[i,j,f_l,f_r]}$	
Goal:	$<S\downarrow \mid \downarrow S>_{\alpha[0,n,-,-]}$	
Axioms:		
Anchor	$<\Gamma_l \mid \Gamma_r>_{\alpha[i,i+1,-,-]}$ $\quad Anchor(\alpha) = w_i$	Γ_l, Γ_r connected walks of α
co-Anchor	$<w_i\downarrow \mid \downarrow w_i>_{unrooted[i,i+1,-,-]}$	
Rules:	(for left expansion, right is symetrical)	
Substitution	$\dfrac{<X\downarrow \mid \downarrow X>_{\alpha'[i,j,f'_l,f'_r]}<\downarrow X\Gamma_l \mid \Gamma_r>_{\alpha[j,k,f_l,f_r]}}{<\Gamma_l \mid \Gamma_r>_\alpha [i,k,f_l \oplus f'_l, f_r \oplus f'_r]}$	**wrap-3** and co-Anchor recognition
Right Adjunction	$\dfrac{<{}^{*}rX_{(foot)}\Gamma'_l \mid rX^*_{(foot)}>_{\beta[i,j,-,-]}<rX^*_{(\eta)}\gamma lX^*_{(\eta)}\Gamma_l \mid \Gamma_r>_{\alpha[j,k,f_l,f_r]}}{<\gamma\Gamma'_l\Gamma_l \mid \Gamma_r>_{\alpha[i,k,f_l,f_r]}}$	$\Gamma'_l \in \{lX^*, \$\}$
Left Adjunction	$\dfrac{<{}^{*}lX_{foot} \mid lX^*_{foot}>_{\beta[i,j_l,-,-]}<lX^*_{(\eta)}\Gamma_l \mid \Gamma_r>_{\alpha[j_l,k,f_l,f_r]}}{<\Gamma_l \mid \Gamma_r>_{\alpha[i,k,f_l,f_r]}}$	$\eta \notin$ sl spine(α)
Left Adj on spine	$\dfrac{<{}^{*}lX_{foot} \mid \Gamma'_r lX^*_{foot}>_{\beta[i,j,-,-]}<lX^*_{(\eta)}\Gamma_l \mid \Gamma^*_r rX_{(\eta)}>_{\alpha[j,k,f_l,f_r]}}{<\Gamma_l \mid \Gamma_r\Gamma'_r>_{\alpha[j,k,f_l,f_r]}}$	$\Gamma'_r \in \{{}^{*}rX, \$\}$
Sub-tree extraction	$\dfrac{<{}^{*}lX \mid rX^*>_{\beta[i,j,j_l,j_r]}<rX^*_{(\eta)}\gamma lX^*_{(\eta)}\Gamma_l \mid \Gamma_r>_{\alpha[j,k,f_r,f_l]}}{<\gamma lX^*_{(\eta)}X\eta\downarrow \mid \downarrow X\eta rX^*_{(\eta)}>_{\alpha[j_r,j_r,-,-]}<\downarrow X\eta\Gamma_l \mid \Gamma_r>_{\alpha[j,k,f_r,f_l]}>}$	**wrap-1**
Wrap Adj on spine	$\dfrac{<{}^{*}lX \mid rX^*>_{\beta[i,j,j_l,j_r]}<rX^*_{(\eta)}\Gamma_l \mid \Gamma_r{}^{*}lX_{(\eta)}>_{\alpha[j_l,j_r,f'_l,f'_r]}}{<\Gamma_l \mid \Gamma_r>_{\alpha[i,j,f'_l,f'_r]}}$	**wrap-2** adjunction on sub-tree
No Left Adjunction	$\dfrac{<lX^*_{(\eta)}\Gamma_l \mid \Gamma_r>_{\alpha[i,j,f_l,f_r]}}{<\Gamma_l \mid \Gamma_r>_{[i,j,f_l,f_r]}}$	
Gap creation	$\dfrac{<{}^{*}lX_{(foot)}\Gamma_l \mid \Gamma_r>_{\beta[i,f_r,-,-]}}{<\Gamma_l{}^{*}rX_{(foot)} \mid lX^*_{(foot)}\Gamma_r>,j]_{\beta[i,j,f_r,j]}}$	$\Gamma_l \notin \{{}^{*}lX, \$\}$

Table 1: Deductive system for an LTAG bidirectional chart-parser, lexically guided and based majoritarily on trees, thanks to a precompilation of their nodes into left and right walks.

The active connector is poped on extreme left (resp. right) of its stack Γ_l (resp. Γ_r). Each connector is associated with its node η, though we do not always mark it. The *spine* is the path from anchor to root.**wrap-1**, **wrap-2**, **wrap-3** identifie the three steps of a wrapping adjunction on an internal node, cf. figure 1.

estimates the pruning model. Types are described in section 2, their use in the deduction system, in section 3, their use for item-pruning in section 4.

2 Linear Typing

Guiding the Tree expansion We guide the parsing by independent left and right *connected-walks*, inspired from (Lavelli and Satta, 1991) bidirectional parser and (Lopez, 1998) connected routes. Left and right connected walks follow respectively left- and right- monotonic expansion, outward, from the anchor to the root, as displayed in figure 2. They list node operations considered as *connectors*.

Link-Grammar expression To express linear types, we exploit the Link-Grammar formalism (Lafferty, Sleator, and Temperley, 1992), which is close to Categorial Grammar. Left and right walks are expressed as stacks of connectors, so that the extreme connector is the one to connect the closest to the anchor[1] An illustration is given in table 2 for the tree in figure 2.

Typing strategy. In its own walk, the foot bears the adjunction, with type *l* or *r* inversly to the foot side. In the opposit walk, the foot-node may be reached as well, provided that there is a direct path from root to foot. In the deduction system, in table 1, the foot-node of a left or right auxiliary tree achieves adjunction, but the foot-node of a wrapping auxiliary tree creates a gap and passes its adjunction

[1]The derivation is represented as a fully connected and oriented graph of trees whose edge labels are connector names (provided that a sub-tree is extracted to decompose wrapping adjunction, cf. figure 1.

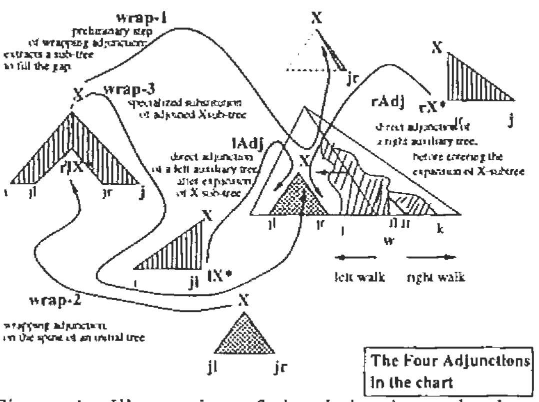

Figure 1: Illustration of the deduction rules in table 1.

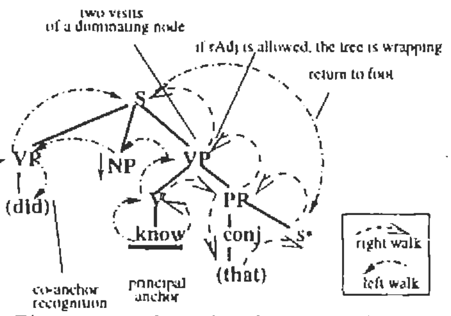

Figure 2: Left and right tree walks.

capacities to the root-node, with an opposit type for the opposit sides.

It can be noted that each node that can receive adjunction yields two linked connectors, which bound the sub-list of connectors of their sub-tree.

3 Deductive Chart Parser

We wish to get elementary-like types on derived structure, so as to use a super-tagging-like model to prune derived paths. We try thus to keep as close as possible to trees when driving thé parsing. But we are not aiming at top-down parsing, since we wish to evaluate derived paths that span the input. This leads to isolating wrapping adjunction from left- and right-, adjunctions, since it is the only case where sub-tree extracting is unavoidable (cf. figure 1). Actually this emphasis on wrapping auxiliary trees is not surprising, since they account for LTAG context-sensitiveness (Schabes and Shieber, 1994).

The full deductive parsing system is defined in table 1. for the LTAG bidirectional-chart parser.

Our approach advantage is threefold: first, it considers only operations that are lexically sound, according to the input string sequence; second, it keeps the number of spurious items very low, by creating very few *sub-tree-* (or *node-*) items; third, it isolates clearly the CF-component, so that the parsing behaves very nicely when faced with near-CF derivations, which are a majority in practice.

left walk meta-rule:
$^{*}S_{foot}$ S^* NP^* $\downarrow$did $_r VP^*$ NP$\downarrow$ NP^* $N^*\leftarrow\bullet$ know

right walk meta-rule:
know $\bullet\rightarrow$ $^{*}_{r}V$ that$\downarrow$ S^*_{foot} (NA$^{*}_{r}$VP) $^{*}_{r}S$

left and right connector stacks:
$<N^* \cdots S^*$ $^{*}S_{foot}$ | $^{*}_{r}S \cdots ^{*}_{r}V>$

type: abstraction on connector stacks, removes specialized substitutions:
co-Anchor: $w\downarrow \rightarrow$ LEX$\downarrow$ sub-tree: $X\eta\downarrow \rightarrow X\downarrow$

Table 2: Typing the tree in figure 2.
In a right walk, lX^* expresses an auxiliary root-node and $^{*}lX$, a node expecting adjunction, $X\downarrow$ expresses a substitution site and $\downarrow X$, the root of an initial tree. In a left walk they work the other way around.

Now, regarding complexity, first three "near-CF" rules yield a worst-case complexity of $O(n^5)$, wrapping adjunction on a lexical spine yields $O(n^6)$, but the sub-tree rule yields $O(n^7)$. We could change that rule into a "systematic" snb-tree extraction with arbitrary gap frontiers, in order to go down to $O(n^5)$, bnt this would generate a lot of spnrious items. Therefore we prefer a lexical check with a wrapping auxiliary tree, since their occurrence is marginal.

4 Probabilistic Thresholding

Probabilistic parsing is expressed through the deductive system as follows:

$$\begin{array}{ll} [_i item_j] & = P_{\mathcal{I}}(_i item_j) = P(item \stackrel{*}{\Rightarrow} w_i \ldots w_j) \\ Rule & = P(rule) \end{array}$$

$$\frac{Rule, [_i item_j][_j item_k]}{[_i item_k] = Rule * [_i item_j] * [_j item_k]} \; d$$

Probabilities of items are *inside probabilities* i.e generative probability that an item dominates its current span of input. Now the usefulness of an item in reaching full derivations is mainly in the *outside probability* $P_{\mathcal{O}}$ of that item, defined for LTAG in (1), following (Schabes, 1992)

$$\begin{array}{c} \text{for } pos = [i, j, f_l, f_r] \\ P\mathcal{O}([s]_{pos}) = \sum_{U,V,T} P([S] \stackrel{*}{\Rightarrow} U[s]_{pos} VT) \\ s.t. U \stackrel{*}{\Rightarrow} \mathcal{W}_{0,i}, V \stackrel{*}{\Rightarrow} \mathcal{W}_{j,n}, T \stackrel{*}{\Rightarrow} \mathcal{W}_{f_l,f_r} \end{array} \quad (1)$$

$$\begin{array}{c} P_{prior}([s]) = \sum_{U,V,T,\mathcal{W}'} P(S \stackrel{*}{\Rightarrow} U[s]VT) \\ s.t. UV[s]T \stackrel{*}{\Rightarrow} \mathcal{W}' \end{array} \quad (2)$$

$$\tilde{P}([s]_{pos}) = \sum_{U_m, V_p, T_q} P(S \stackrel{*}{\Rightarrow} U_{1..m} V_{1..p} [s]_{pos} T_{1..q}) \quad s.t. U_{1..m} V_{1..p} [s]_{pos} T_{1..q} \stackrel{*}{\Rightarrow} W \tag{3}$$

For an item-path, outside probability accounts for parsing-deductions to come, i.e the connectors of the item stacks. Whereas consumed connectors are responsible for the inside probability. There is no way to compute the outside probability without the knowledge of the actual "connection" of connectors, but this decomposition gives us a very precious means to normalize inside probabilities, which put very low probabilities on large items.

item-path	$U = (U_1, \ldots, U_p)$
remaining stacks:	$< \Gamma_i^l \mid \Gamma_i^r >$
consummed stacks:	$< \Gamma_i^{l\prime} \mid \Gamma_i^{r\prime} >$
$P_I(U) \approx$	$\sqrt{\prod_i \Gamma_i^{l\prime} \Gamma_i^{r\prime} prod_i P_0^i}$
$P_O(U) \approx$	$\sqrt{\prod_i \Gamma_i^l \Gamma_i^r}$

(Goodman, 1998) proposes two useful approximation of the outside score of item[s], in order to correct the inside probability bias. We express them in the context of LTAG in (2) and (3). The first one simply corresponds to the prior probability of the item category. The second one is the cumulated probability of all item-paths $U = (U_1, .., U_p)$ that include [s]. This value can be computed in several passes (Goodman, 1998; Rayner and Carter, 1996).

Computing path probabilities entails estimating the probability that sequences of items, which span the input, can build a complete derivation. This is the aim of Super-tagging, which can be viewed as a model for the first step of the chart. We generalize it to model steps n, i.e a step where edges have maximal length n. Here are some approximations which have been proposed:

$$\begin{array}{lcll} P_I(U) & = & P(U_1..U_p) & \text{Real} \\ & \approx & \prod_i P(U_i) & \text{Fully independent} \\ & \approx & \prod_i P(U_i \mid U_{i-1}) & \text{Markovian} \\ & \approx & Min_i min(P(U_i \mid U_{i-1}), & \text{Fully dependent} \\ & & P_{prior}(U_i), P(U_i \mid U_{i+1})) & \end{array}$$

Item sequences ressemble elementary tree sequences, as they share types, and connect through the same connectors (provided the type abstraction explained in table 2 for specialized substitutions). Hence the possible re-use, in a first approximation, of supertagging training for the generalized item-model.

Smoothing: types decompose into a very small set of connectors, with straigtforward interpretation. They can serve as a useful basis for compnting backoff probabilities. For instance by distributing the probability mass of each connection among all types that allow this connection, in the same way as (Lafferty, Sleator, and Temperley, 1992).

5 Discussion

We have presented a general framework for deductive parsing, probabilistic parsing and super-tagging. This unified approach opens a lot of perspective in the design of efficient and robust LTAG parsing. However, it remains to be fully validated.

As far as supper-tagging is concerned, supertags should perform better than linear types as their definition integrates a large amount of linguistic knowledge. Types nonetheless provide for that task a very simple, and yet relevant, smoothing scheme. As for further steps of parsing, types turn out very adequate, as they allow to express in a simple manner the essential computations involved.

References

Caroll, John and David Weir. 1997. Encoding frequency information in lexicalized grammars. In *ACL/SIGPARSE workshop on Parsing Technologies*, MIT, Cambridge.

Goodman, J. 1998. *Parsing Inside-Outside*. Ph-D, Division of Engineering and Applied Sciences, Harvard University, May.

Joshi, A. and B. Srinivas. 1994. Disambiguation of Super Parts of Speech (or Supertags): Almost Parsing. In *COLING'94*, Kyoto, Japan, August.

Lafferty, J., D. Sleator, and D. Temperley. 1992. Grammatical trigrams: A probabilistic model of link grammar. In *Symposium on Probabilistic Approaches to Natural Language*. AAAI.

Lavelli, A. and G. Satta. 1991. Bidirectional parsing of Lexicalized Tree Adjoining Grammars. In *EACL'91*.

Lopez, Patrice. 1998. Analyse guidée par connexité des TAG lexicalisées. In *TALN'98*, Paris, France.

Rayner, Manny and David Carter. 1996. Fast parsing using pruning and grammar specialization. In *ACL'96*.

Schabes, Y. 1992. Stochastic Lexicalized Tree-Adjoining Grammars. In *COLING'92*, Nantes, France, August.

Schabes, Y. and S. Shieber. 1994. An alternative conception of tree-adjoining derivation. *Computational Linguistics*, 20(1):91–124.

Shieber, Stuart M., Yves Schabes, and Fernando C. N. Pereira. 1994. Principles and implementation of deductive parsing.

van Noord, G. 1994. Head-corner parsing for TAG. *Computational Intelligence*, 10(4):525–534.