Exploring the Spinal-Tig Model for Parsing French

Djamé Seddah

Equipe Projet Alpage (Inria) & Université Paris-Sorbonne
28 rue Serpente
F-75006 Paris — France
djame.seddah@paris-sorbonne.fr

Abstract

We evaluate statistical parsing of French using two probabilistic models derived from the Tree Adjoining Grammar framework: a Stochastic Tree Insertion Grammar model (STIG) and a specific instance of this formalism, called Spinal Tree Insertion Grammar model which exhibits interesting properties with regard to data sparseness issues common to small treebanks such as the Paris 7 French Treebank. Using David Chiang's STIG parser (Chiang, 2003), we present results of various experiments we conducted to explore those models for French parsing. The grammar induction makes use of a head percolation table suitable for the French Treebank and which is provided in this paper. Using two evaluation metrics, we found that the parsing performance of a STIG model is tied to the size of the underlying Tree Insertion Grammar, with a more compact grammar, a spinal STIG, outperforming a genuine STIG. We finally note that a spinal framework seems to emerge in the literature. Indeed, the use of vertical grammars such as Spinal STIG instead of horizontal grammars such as PCFGs, afflicted with well known data sparseness issues, seems to be a promising path toward better parsing performance.

1. Introduction

The use of Tree Adjoining Grammar-based formalisms (Joshi, 1987), henceforth TAG, for probabilistic parsing is not a new idea. While the initial intuition of adding probabilities to TAGs was mentioned by Joshi et al. (1975) in their seminal paper, formulations of a TAG probabilistic model were independently proposed in (Schabes, 1992) and (Resnik, 1992). Schabes (1992) proposed a version of the Inside-Outside algorithm for reestimating a Stochastic Lexicalized TAG (LTAG). This algorithm was later used by Hwa (1998) for the unsupervised learning, from partially bracketed data, of a context free variant of TAGs, namely Lexicalized Tree Insertion Grammars (LTIG, (Schabes and Waters, 1995)). Using LTIG allows for the same parsing complexity as pure context-free grammars, while permitting the generation of richer derivational structures.

However, none of these works presented the extraction of linguistically motivated TAGs. Indeed, such grammars can be induced from a set of parses using hand written heuristics, such as head percolation and argument adjunct distinction tables, for reconstructing the derivations. This approach was, for instance, followed by (Neumann, 1998; Xia, 1999; Chen and Vijay-Shanker, 2000; Chiang, 2000) to extract various kinds of lexicalized tree grammars to build statistical parsers, supertaggers and so on.

Interestingly, another operation, the *sister-adjunction*, is added to the Stochastic TIG (STIG) framework by Chiang (2000) in order to handle flat structures inherent to most treebanks. Furthermore, a special mode of the Chiang's STIG model implementation can lead to the extraction of LTIGs that only contain *spines* (i.e. lineage from the anchor to its maximal projection), allowing for more compact grammars that can counterbalance data sparseness issues in a highly lexicalized formalism such as LTIG. We call this model SPINAL STIG and we present here an evaluation of its performance on the Paris 7 French Treebank (FTB, (Abeillé et al., 2003)) via the use of the Chiang (2003) STIG parser.

This paper is structured as follows: we provide a brief overview of the FTB, then we introduce the STIG and SPINAL STIG models and outline their extraction process. We present various possible parameter setups affecting the spinal grammar probabilistic model. We then provide a comparison between pure STIG and SPINAL STIG models run on different instances of the FTB. Before concluding, we discuss our results compared to previous works.

2. The French Treebank

Unlike the Penn Treebank (Marcus et al., 1994), the FTB proposes a flat annotation scheme (Abeillé et al., 2003). For instance, there are no VPs for finite verbs and only one sentential level for clauses or sentences regardless of whether or not they are introduced by a complementizer. Only the *verbal nucleus* (VN) is annotated and comprises the verb, its clitics, auxiliaries, adverbs and surrounding negation. Because of this lack of VP constituent for finite verbs in both treebanks, the distinction between argument and adjunct is not marked by a different structure but rather by functional annotations. We use the treebank variation described in (Crabbé and Candito, 2008) with the CC tagset. All compounds are explicitly marked. We refer to this treebank instance as FTB or FTB-CC wherever the distinction is relevant.

3. Overview of Lexicalized Tree Adjoining Grammars

A Lexicalized TAG grammar (LTAG) consists of a large lexicon where each lexical entry (*i.e.,* anchor) is associated with a set of elementary trees.

Two composition operations are provided: the substitution which is a context free derivation of an elementary tree to a given leaf node of any tree, and the adjunction which inserts a particular kind of tree on a node. The tree being adjoined has one special leaf node, marked by an asterisk and with the same label as the root node of the adjoined tree. Those trees are called *auxiliary* trees whereas non-auxiliary elementary trees are called *initial* trees.

As the formalism's basic units are trees, the derivation tree and the derived tree (*i.e.,* the parse tree) are not isomorphic. The derivation tree strictly records the resulting operation leading to a derived tree. Each node of the derivation tree is labeled by a Gorn address[1] telling us where the operation took place in the dominating tree.

A toy grammar and a sketch of derivations for the sentence *Jean aime beaucoup Marie/John really likes Mary* is shown in Figure 1. Resulting derived and derivation trees[2] are shown in Figure 2.

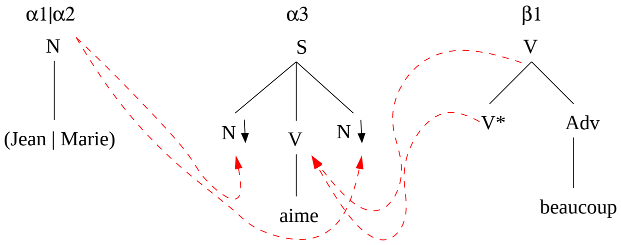


Figure 1: LTAG toy grammar and sketch of derivations for *Jean aime beaucoup Marie/John really likes Mary*

Regarding context free variants of TAG such as TIG, the main differences is that wrapping adjunction of auxiliary trees is not allowed (Schabes and Waters, 1995). This ensures that the weak generative power is strictly the same as in context free grammars and allows for a formalism parsable in cubic time.

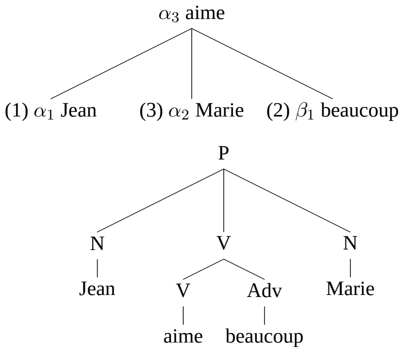


Figure 2: Derivation tree and derived tree of "Jean aime beaucoup Marie"

4. Statistical TIG models

In this section, we introduce the Stochastic Tree Insertion Grammar (STIG) variant, introduced by Chiang (2000).

4.1. STIG

STIG is a tree rewriting system with three types of elementary trees: initial, predicative auxiliary trees and three composition operations: substitution, adjunction and sister-adjunction. While the first two are well known, being the classical TAG operations introduced above, the sister-adjunction was introduced in (Rambow et al., 1995) and used by Chiang (2000) as a means to derive treebank flat structures. It is a context-free operation only constrained through a probability model that conditions the generation of a modifier tree on a given node upon the root label of the previously generated sister-adjoining tree.

As opposed to the induction of CFGs that can be easily extracted from a treebank, the reconstruction of STIG derivations must rely on heuristics stating which nodes will be part of an elementary tree. Following previous works on lexicalized PCFG induction (e.g. (Magerman, 1995; Collins, 1997)), the Chiang's (2000) model makes use of a head-percolation table to distinguish the path from a lexical anchor to its maximal projection. This path is called *spine*.[3]

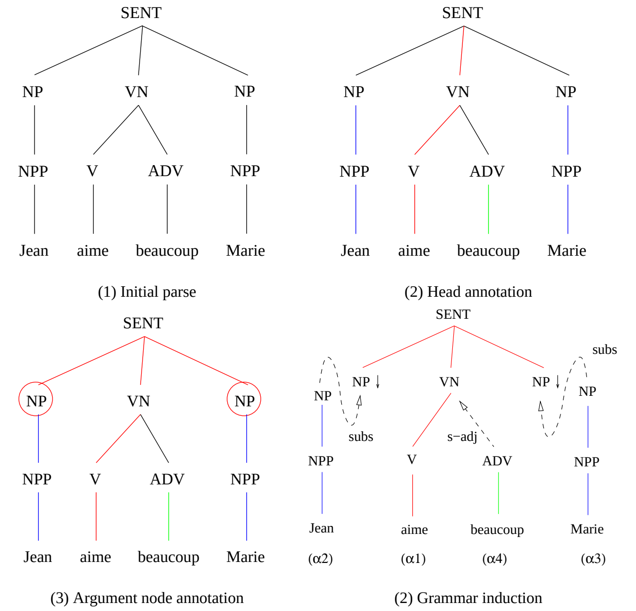
In the LTAG framework, all elementary trees are extended projections of lexical items and contain

[1] The root has the address 0, the i^{th} child of the root has address i and for all other nodes: the i^{th} child of the node with address j has address $j.i$.

[2] For the sake of simplicity, each tree whose name begins with β, resp. α, is governed by adjunctions, resp. substitution.

[3] Note that in the TAG framework, the term *spine* usually denotes the path from a root node of an auxiliary tree to its foot node.

all syntactic arguments of a lexical anchor (*i.e.*, the syntactic head). These argument nodes correspond to substitution nodes of elementary trees and in order to properly annotate these nodes, an argument-adjunct distinction table is required (Xia, 1999). Given the non-configurational nature of the French treebank, our rules are based on functional annotations. In Figure 3, we outline the STIG induction process from a parse tree based on the FTB's annotation scheme (1) to the STIG grammar underlying the reconstructed derivations (4).



(1) Initial parse

(2) Head annotation

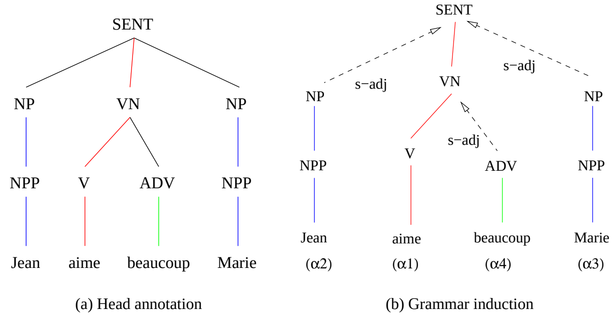
(3) Argument node annotation

(2) Grammar induction

Figure 3: STIG extraction process
(1) Jean aime beaucoup Marie *'John really likes Mary'*

4.2. Spinal STIG

As shown in section 4.1., the conjunction of head-percolation and argument/adjunct rules is essential to extract proper elementary trees from treebank data. Interestingly, having no access to an argument adjunct table leads the Chiang's (2000) STIG model implementation[4] to extract a grammar where all extracted trees have no argument nodes. All of these trees are therefore made of *spines* and consequently handled by sister-adjunction. This behavior results from an undocumented side effect of the Chiang (2000) model and was noticed during its preliminary adaptation to French. Figure 4 shows the extraction process of such a spinal grammar.
We notice that even though the extracted grammars are different and lead to different derivation trees[5], they share the same dependency structure (Fig. 5).



(a) Head annotation

(b) Grammar induction

Figure 4: Spinal STIG extraction process for (1)

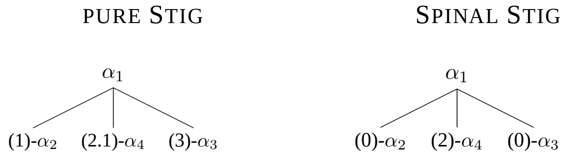


Figure 5: Derivation trees for (1)

Finally, in order to ease the estimation of the probability model's parameters, the grammar is implicitly split between tree templates and lexical anchors, allowing the probabilistic model to generate first a template, then its anchor (Chiang, 2000). We describe the parameter classes of the STIG models in the next section.

4.3. STIG/STAG Parameter classes

Because TIGs are a subset of TAGs that retains the same properties, regardless on constraints used to avoid the generation of wrapping auxiliary trees, a probabilistic model for Stochastic TAGs is also valid for Stochastic TIGs. A Stochastic TAG (Resnik, 1992; Schabes, 1992), henceforth STAG, is an history based model, such as PCFGs, where the probability of a TAG derivation tree is the product of the probabilities of the operations used for its construction.
Chiang (2003) describes the parameters of a STAG as follows:

$$\sum_\alpha P_i(\alpha) = 1$$

$$\sum_\alpha P_s(\alpha|\eta) = 1$$

$$\sum_\beta P_a(\beta|\eta) + P_a(NONE|\eta) = 1$$

Following the usage in the TAG literature, α denotes initial trees, β auxiliary trees, γ any tree and η a node of a tree. $P_i(\alpha)$ is the probability of a derivation starting by α, $P_s(\alpha|\eta)$, the probability of a substitution of α on η, $P_a(\beta|\eta)$, the probability of β adjoining on η and $P_a(NONE|\eta)$, the probability of having zero adjunction on η.

[4] Available at www.isi.edu/∼chiang/software/hybrid.tar.gz

[5] With respect to the types of the derivations and their respective site nodes.

As we mentioned, the sister-adjunction operation was added by Chiang (2000) to the original STAG model in order to cope with treebanks' flat annotation scheme. Originating from D-tree grammars (Rambow et al., 1995), this operation allows for any initial tree to be inserted as a new daughter of a node. The sister-adjunction also bears some resemblances with the *furcation* operation as described in (Halber, 1998; Kilger and Poller, 2000). Its capacity to derive multiple initial trees in the same node is also reminiscent of the *modifier adjunction* of Schabes and Shieber (1994).

Its parameter class is the following (Chiang, 2003):

$$\sum_{\alpha} P_{sa}(\alpha|\eta, i, X) + P_{sa}(STOP|\eta, i, X) = 1$$

Where α denotes initial trees and (η, i) varies over possible sister-adjunction sites. $P_{sa}(\alpha)$ the probability of sister-adjoining α and $P_{sa}(STOP)$ the probability of no further sister-adjunction. X is the root label of the previous tree that was closer to the head, to sister-adjoin at the site node (η, i).

The backoff structures of this model are presented in the Appendix, Table 4.

5. Head rules and argument-adjunct distinction tables

All lexicalized parsers use head propagation tables. Adapting them to the French language requires to design French specific head propagation rules. To this end, we used those described by (Dybro-Johansen, 2004) for extracting a Tree Adjoining Grammar from an early release of the FTB. From this set, we built a set of meta-rules that were automatically derived to match each treebank annotation scheme that will be used in the next two sections. Table 5 (Appendix) provides an example of such a head rule percolation table for the FTB-CC.

As the pure STIG model needs to distinguish between argument and adjunct nodes to extract proper initial trees, we implemented an argument-adjunct distinction table that takes advantage of the function labels annotated in the treebank.

6. Varying the generating context of the spinal trees

As shown in section 4.3., the probability of an initial tree to sister-adjoin on a site node (η, i) is conditioned on a context X, where X is set by default to the root label of the previous tree that was sister-adjoined on this site node.

As this parsing model allows for modifying this setup, and because when run in spinal mode, all trees of the extracted grammar are handled by the sister-adjunction, it is interesting to observe the impact of this context on parsing performance.

Thus, in this section we present results from experiments where the context is either a flag stating that the tree to be generated is the first modifier tree to be sister-adjoined (first), the root label of the previous sister¯adjoined tree (root) or no context at all (none).

We also test whether generating the modifier trees from the head outward instead of left-to-right has an impact on a treebank as flat as the FTB. The idea is to verify if the parsing of a relatively free word order language such as French [6] can benefit from a generative process less dependent upon a recorded phrase order.

All evaluations are given from the SPINAL STIG parsing of the canonical development section of the FTB-CC with gold part-of-speech supplied for unknown words. The evaluation metrics is PARSEVAL's labeled brackets and POS tagging accuracy (Black et al., 1991). As usual, scores are given for sentences of length less than 41 words.

Results are presented in Table 1 and show that the default setup conditioning the generation of spinal trees upon the previous root label provides by far the best performance when the heads are generated from left to right. Nonetheless, the situation is reversed when the trees are generated from the head outward.[7] One can notice the huge gap between the POS tagging accuracies in the two parts of Table 1. The point is that generating modifier trees from left to right means that their root labels are generated by a first-order Markov process (Chiang, 2003). Thus, we can hypothesize that this process also applies to the label of their pre-terminal nodes (the POS). In fact, conditioning POS with a Markovian process respecting the word order is probably more efficient that conditioning them on a maybe more distant tree that has been sister-adjoined to the head regardless of their relative position on the word stream.

In the remaining part of this paper, we use the "root" context with trees generated from left to right.

[6] French exhibits a limited amount of word order variation occurring at different syntactic levels including (i) the word level (*e.g.,* pre or post nominal adjective, pre or post verbal adverbs); (ii) phrase level (*e.g.,* possible alternations between post verbal NPs and PPs).

[7] Note that generating modifier trees from the head outward is similar to the way the Collins' model 2 handles modifier non terminal nodes (Bikel, 2004).

context	Rec.	Prec.	F_1	Pos acc.
head outward				
none	71.39	71.32	71.36	**97.27**
first	**79.63**	**78.43**	**79.03**	97.24
root	78.75	77.38	78.06	97.08
left-to-right				
none	71.39	71.32	71.36	97.27
first	81.72	81.59	81.66	97.73
root	**82.9**	**83.13**	**83.01**	**97.91**

Table 1: Sister-adjunction context variation for the Spinal Stig model on the Ftb-cc dev section with unknown word POS supplied

7. Grammar size and parsing performance

In (Seddah et al., 2009), a benchmarking between various probabilistic models applied to French treebanks showed that the parsing performance of the spinal STIG models ranks roughly in the middle of lexicalized parsers w.r.t to the classical labeled bracket metrics when evaluated on the Ftb. However, it reaches the state of the art regarding unlabeled dependency evaluation.

Here we extend those results by showing that the parsing model performance is bound to the size of the extracted grammar as shown by Table 3.

The same experimental protocol[8] as Seddah et al. (2009) is used on three instances of the Ftb with different tagsets (Table 2).

The tagset min contains only bare syntactic categories, the tagset cc is the one described in (Crabbé and Candito, 2008) and known to provide the best parsing performance on the Ftb, the tagset schlu is the one extracted from the Modified French Treebank (Schluter and van Genabith, 2007), henceforth MFT.[9]

tagset	POS
min	A ADV C CL D ET I N P P+D P+PRO PONCT PREF PRO V
cc	ADJ ADJWH ADV ADVWH CC CLO CLR CLS CS DET DETWH ET I NC NPP P P+D P+PRO PONCT PREF PRO PROREL PROWH V VIMP VINF VPP VPR VS
schlu	A A_card ADV ADV_int ADVne A_int CC CL C_S D D_card ET I N N_card P P+D PONCT P+PRO_rel PREF PRO PRO_card PRO_int PRO_rel V_finite V_inf V_part

Table 2: Ftb POS Tagsets

Jointly with the ParsEval F_1 labeled bracket metric, we provide results of unlabeled dependency evaluation which is computed using the algorithm described in (Lin, 1995). In order to provide a realistic view of the performance of Stig models, the different Ftb test sets are here tagged with the TnT tagger of Brants (2000) trained on the training section of each instance of the Ftb.

tag set	min	CC	shlu
pure Stig			
# of tree templates	428	**414**	633
Labeled Bracket F$_1$	80.52	**81.18**	81.16
Unl. Dep. F$_1$	87.98	**88.67**	87.08
Spinal Stig			
# of tree templates	139	**83**	104
Labeled Bracket F$_1$	80.66	**81.73**	81.54
Unl. Dep. F$_1$	87.92	88.85	**89.02**

Table 3: pure and spinal Stig parsing evaluation results on the Ftb test set with different tag sets

Results (Table 3) show that in all cases, a Spinal Stig model outperforms a genuine Stig using constituency metric, the most compact grammar offering the best performance. However, unlabeled dependencies results are somehow a bit higher on the Ftb-Schlu (*i.e.*, the Ftb with the schlu tagset) with Spinal Stig parsing. This tagset was conceived to optimize LFG induction from the Mft and contains more morphological information than the cc tagset. This supplementary amount of data brought to the head-percolation table used for the Ftb-Schlu parsing leads the Lin (1995) dependency extraction process to be less error-prone. This can explain this small difference in these results.

We also note that the best results are provided by the more compact grammars. These grammars are extracted from the Ftb-cc. This shows that the cc tagset granularity seems to be an advantage for STIG parsing.

8. Related work and discussion

In (Sangati and Zuidema, 2009), a process of automatically inducing a Lexicalized Tree Subtitution Grammar from a treebank is described. Their goal is to validate the automatic induction of head rules from annotated data. In their process, the first extracted grammar is actually a spinal STIG. Also, note that a spinal version of a PCFG is briefly described in (Post and Gildea, 2009) in order to test the performance of different kinds of binarized grammars. In this approach, each spinal tree corresponds to a set of CFG rules, the

[8] The first 10% of the Ftb are used as the test set, the next 10% for the development set and the rest for training (*i.e.*, 1235/1235/9881 sentences).

[9] See (Seddah et al., 2009) for further details on these tagsets and their relative parsing performances.

root node being considered as a left hand-side non-terminal symbol.

Closer to our goals, a LTAG SPINAL model has been proposed by Shen and Joshi (2005) and developed in (Shen et al., 2008). Their grammar extraction algorithm takes advantage of the PropBank annotations (Palmer et al., 2005) in order to provide much deeper syntactic dependencies than the ones underlying our spinal grammars based only on a simple set of heuristics. Note that the extracted LTAG SPINAL grammar is one order of magnitude less coarse than our own, as it contains 1,224 spinal trees.

Recently, a similar though richer model has been independently proposed in (Carreras et al., 2008) with the purpose of getting a powerful and very sophisticated parsing model that can use non local features. As these models are closely related, we ran the spinal TIG model on the section 23 of the Penn Treebank and obtained 87.79% of PARSEVAL labeled bracket F_1 score with 369 tree templates[10] while Carreras et al. (2008) report very high state-of-the-art results (91.1%). This can be explained by the use of high level machine learning techniques to cope with spine attachment decisions whereas our model is based on a pure generative model.

As opposed to some of the works we briefly presented, we did not try to refine the spinal grammars we extracted, for example we did not discriminate between spinal trees sister-adjoining on the right and those sister-adjoining on the left of an elementary tree. The idea for us was to see how far we could get using the smallest possible grammar we could induce from a treebank. Indeed, we showed that the parsing performance for French was tied to the grammar size. More compact grammars in the STIG model lead to better results.

We believe that refining the model to take into account more context and to split the grammar according to the most probable side of sister-adjoining trees will improve the parsing performance of the SPINAL STIG model. Nonetheless, this is not the main point of dealing with spinal grammars. We think that going from horizontal grammars such as PCFGs to vertical ones as implied by the emerging SPINAL framework is currently one of the best ways to deal with the unavoidable data sparseness issues due to the current average limited size of usual treebanks.

9. Conclusion

We reported evaluation parsing results from the extraction of various STIG for French. We showed that the SPINAL STIG model outperforms a genuine STIG model when applied on a less hierarchical treebank such as the FTB. We showed that having more compact grammars leads to better parsing performance for parsing French in a TAG framework. In our future work, we will refine the extraction model in order to extract less coarse grammars.

Acknowledgments

This work was supported by the ANR Sequoia (ANR-08-EMER-013). We are grateful to our anonymous reviewers for their comments and to David Chiang for making his parser freely available. We thank Benoit Crabbé and Marie Candito for our many discussions on this topic. All remaining errors are ours.

10. References

Anne Abeillé, Lionel Clément, and François Toussenel, 2003. *Building a Treebank for French*. Kluwer, Dordrecht.

Daniel M. Bikel. 2004. Intricacies of Collins' Parsing Model. *Computational Linguistics*, 30(4):479–511.

Ezra Black, Steve Abney, Dan Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, Fred Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of english grammars. In *Proceedings of the 1991 DARPA Speech and Natural Language Workshop*, pages 306–311.

Thorsten Brants. 2000. Tnt – a statistical part-of-speech tagger. In *Proceedings of the 6th Applied NLP Conference (ANLP)*, Seattle-WA.

Xavier Carreras, Mickael Collins, and Terry Koo. 2008. TAG, dynamic programming, and the perceptron for efficient, feature-rich parsing. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning (CoNLL)*, pages 9–16.

John Chen and K. Vijay-Shanker. 2000. Automated extraction of tags from the Penn TreeBank. In *Proceedings of the 6th International Workshop on Parsing Technologies (IWPT'06)*, pages 65–76. Springer.

David Chiang. 2000. Statistical parsing with an automatically-extracted Tree Adjoining Grammar. *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 456–463.

[10] On the same PTB section, we obtained 88.69% for the PURE STIG model with 868 templates. All scores are given for sentences of length less than 41.

David Chiang, 2003. *Statistical Parsing with an Automatically Extracted Tree Adjoining Grammar*, chapter 16, pages 299–316. CSLI Publications.

Michael Collins. 1997. Three Generative, Lexicalized Models for Statistical Parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 16–23, Madrid, Spain.

Benoit Crabbé and Marie Candito. 2008. Expériences d'analyse syntaxique statistique du français. In *Actes de la 15ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN'08)*, pages 45–54, Avignon, France.

Ane Dybro-Johansen. 2004. Extraction automatique de Grammaires d'Arbres Adjoints à partir d'un corpus arboré du français. Master's thesis, Université Paris 7.

Ariane Halber. 1998. Tree Grammars Linear Typing for unified Super-Tagging/Probabilistic Parsing Model. In *Proceedings of the 4^{th} International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+4)*, Philadelphia, USA.

Rebecca Hwa. 1998. An empirical evaluation of probabilistic Lexicalized Tree Insertion Grammars. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, volume 36, pages 557–563, Montreal, Canada, August.

Aravind K. Joshi, Leo S. Levi, and Masako Takahashi. 1975. Tree Adjunct Grammars. *Journal of the Computer and System Sciences*.

Aravind K. Joshi. 1987. Introduction to Tree Adjoining Grammar. In A. Manaster-Ramer, editor, *The Mathematics of Language*. J. Benjamins.

Anne Kilger and Peter Poller. 2000. Cdl–tags: A grammar formalism for flexible and efficient syntactic generation. In *Proceedings of the 5^{th} International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+5)*, volume 5, pages 43–48, Paris, France.

Dekang Lin. 1995. A dependency-based method for evaluating broad-coverage parsers. In *International Joint Conference on Artificial Intelligence*, pages 1420–1425, Montreal.

David M. Magerman. 1995. Statistical decision-tree models for parsing. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 276–283. Association for Computational Linguistics Morristown, NJ, USA.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of English: The Penn TreeBank. *Computational Linguistics*, 19(2):313–330.

Gunter Neumann. 1998. Automatic extraction of Stochastic Lexicalized Tree Grammars from treebanks. In *Proceedings of the 4th Workshop on Tree-Adjoining Grammars and Related Frameworks*, pages 120–123, Philadelpia, Pennsylvania.

Martha Palmer, Paul Kingsbury, and Daniel Gildea. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Matt Post and Daniel Gildea. 2009. Weight pushing and binarization for fixed-grammar parsing. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 89–98, Paris, France, October. Association for Computational Linguistics.

Owen Rambow, K. Vijay Shanker, and David Weir. 1995. D-tree grammars. In *33rd Conference of the Association of Computational Linguistics (ACL'95)*, pages 151–158.

Philip Resnik. 1992. Probabilistic Tree-Adjoining Grammars as a framework for statistic natural language processing. *COLING'92, Nantes, France*.

Federico Sangati and Willem Zuidema. 2009. Unsupervised methods for head assignments. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 701–709, Athens, Greece. Association for Computational Linguistics.

Yves Schabes and Stuart Shieber. 1994. An alternative conception of Tree-Adjoining derivation. *Computational Linguistics*, 20(1):91–124.

Yves Schabes and R. C. Waters. 1995. Tree Insertion Grammar: A Cubic-Time, Parsable Formalism that Lexicalizes Context-Free Grammar without Changing the Trees Produced. *Computational Intelligence*, 21:479–514.

Yves Schabes. 1992. Stochastic Lexicalized Tree Adjoining Grammars. In *Proceedings of the 14th conference on Computational linguistics*, pages 425–432, Nantes, France. Association for Computational Linguistics.

Natalie Schluter and Josef van Genabith. 2007. Preparing, restructuring, and augmenting a French Treebank: Lexicalised parsers or coherent treebanks? In *Proc. of PACLING 07*, Melbourne, Australia.

Djamé Seddah, Marie Candito, and Benoit Crabbé. 2009. Cross parser evaluation and tagset variation: A French Treebank study. In *Proceedings of the 11th Internation Conference on Parsing Technologies (IWPT'09)*, pages 150–161, Paris, France, October. Association for Computational Linguistics.

Libin Shen and Aravind K. Joshi. 2005. Incremental LTAG Parsing. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 811–818, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Libin Shen, Lucas Champollion, and Aravind K. Joshi. 2008. LTAG-Spinal and the Treebank. *Language Resources and Evaluation*, 42(1):1–19.

Fei Xia. 1999. Extracting Tree Adjoining Grammars from bracketed corpora. In *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium (NLPRS-99)*, pages 398–403.

Appendix

	$P_{s,a}(\gamma \mid \cdots)$	$P_{sa}(\gamma \mid \cdots)$	$P_w(\omega \mid \cdots)$
	subst., adj. of γ	sister-adj. of γ	gen. of γ's anchor
0	$\tau_\eta, \omega_\eta, \eta_\eta$	$\tau_\eta, \omega_\eta, \eta_\eta, i, X$	$\tau_\gamma, t_\eta, \omega_\eta, X$
1	τ_η, η_η	$\tau_\eta, \eta_\eta, i, X$	τ_γ, t_η, X
2	$\overline{\tau}_\eta, \eta_\eta$	$\overline{\tau}_\eta, \eta_\eta, i$	τ_γ
3	$\emptyset$	$\emptyset$	t_γ

Table 4: STIG Backoff structure (Chiang, 2003)
τ denotes a tree template, t a POS, eta a node and ω an anchor.
γ is the tree to be generated on the site node η_η of the tree template τ_η, ω_η is the lexical anchor of τ_η, $\overline{\tau}_\eta$ is τ_η stripped from its anchor POS tag t_η and X is the root label of the previous tree to sister-adjoin at the site (η_η, i).
Those backoff structures are combined by linear interpolation, see (Chiang, 2003) for details.

```
(
(S1 (first SENT) )
(PONCT (last *) )
(Sint (last VN) (last AP) (last NP) (last PP) (last VPinf)
(last Ssub) (last VPpart) (last A ADJ ADJWH) (last ADV
ADVWH) )
(VPpart (first VPR VPP) (first VN) )
(SENT (last VN) (last AP) (last NP) (last Srel) (last
VPpart) (last AdP) (last I) (last Ssub) (last VPinf) (last PP)
(last ADV ADVWH) )
(COORD (first CS CC PONCT) )
(AP (last A ADJ ADJWH) (last ET) (last VPP) (last ADV
ADVWH) )
(NP (first NPP PROREL PRO NC PROWH) (first NP)
(first A ADJ ADJWH) (first AP) (first I) (first VPpart) (first
ADV ADVWH) (first AdP) (first ET) (first DETWH DET)
)
(VPinf (first VN) (first VIMP VPR VS VINF V VPP) )
(PP (first P) (first P+D) (first NP P+PRO) )
(Ssub (last VN) (last AP) (last NP) (last PP) (last VPinf)
(last Ssub) (last VPpart) (last A ADJ ADJWH) (last ADV
ADVWH) )
(VN (last VIMP VPR VS VINF V VPP) (last VPinf) )
(Srel (last VN) (last AP) (last NP) )
(AdP (last ADV ADVWH) )
(* (first *) )
)
```

Table 5: Head rule percolation table for the FTB-CC adapted from (Dybro-Johansen, 2004) in the Chiang (2000) format