

On the Affinity of TAG with Projective, Bilexical Dependency Grammar

Tom B.Y. Lai^{1,2} Changning Huang³ and Robert W.P. Luk⁴

City University of Hong Kong¹, Tsinghua University², Microsoft Research Beijing³, Polytechnic University of Hong Kong⁴

1. Introduction

This paper describes a projective, bilexical dependency grammar, and discusses its affinity with TAG. Common features of the two formalisms include a tree-like surface syntactic structure and readiness for a lexicalised treatment. TAG surface structures built from elementary and auxiliary trees by means of substitution and adjunction can correspond to trees consisting entirely of lexical nodes and dependency arcs. Lexical anchors in TAG, a well-motivated notion, can also be accommodated in the dependency grammar formalism, provided it is recognized that the dependent, as well as the governor, can have a vote about the formation of a dependency relation. It is noted, however, that mirroring obligatory adjuncts in TAG in dependency grammar can be problematic.

2. Dependency Analysis

2.1. Projective Dependency Structures

Though not supported by all schools of dependency grammar (Tesnière, 1959), some followers of dependency grammar assume that there is a projective surface or back-bone dependency structure. The theoretical foundation of this tradition can be traced to Gaifman (1965) and Hays (1964), and is summed up in the following well-formedness conditions for dependency structures in Robinson (1970):

- one and only one element is independent;
- all others depend directly on some element;
- no elements depend directly on more than one other;
- if A depends directly on B and some element C intervenes between them (in linear order of string), then C depends directly on A or on B or some other intervening element.

These conditions say, in effect, that conforming dependency structures are representable by trees without crossing branches. Of courses, as in other grammar formalisms that pre-suppose a context-free syntactic structure back-bone, additional linguistic constraints can be incorporated in the formalism by means of various mechanisms, e.g. feature unification.

2.2. Dependency Structures without Phrasal Nodes

In the dependency grammar formalism (Lai and Huang, 1998; Lai and Huang, 2000) discussed in this paper, dependency structures are trees consisting entirely of *lexical* nodes. For example, the dependency *tree* for (1a), taken from Abeillé (1993), is (1b):

- (1) a. Jean dort beaucoup
Jean sleeps much
'Jean sleeps a lot.'
- b.
- | | | | |
|-------|------|------|----------|
| | | dort | |
| | | | |
| ----- | | | |
| | subj | | adjunct |
| | | | |
| Jean | | | beaucoup |

When a coarser degree of granularity is warranted by the situation, the actual lexical items in the tree nodes can be replaced by their syntactic categories.

2.3. Statistical Dependency Analysis

In the computational linguistics community, dependency structures are often parsed, exploiting their affinity with phrase-structure structures, with the help techniques used with context-free grammars (Hellwig, 1986). On the other hand, Collins (1996) uses ‘bilexical’ co-occurrence probabilities (of governors and their head daughters) to estimate the likelihood of phrase-structures in the syntactic analysis of sentences. In a recent effort on Chinese (Lai *et al.*, 2001), bilexical probabilities have been used directly to derive, without direct reference to context-free grammar and phrasal structures, dependency structures using a CYK-like algorithm (Eisner, 1996; Eisner, 2000).

The probabilistic model in Lai and Huang (2000) uses conditional probabilities defined in terms of dependencies. Factors considered include both dominance and ‘function’, as well as other contextual factors like relative proximity to the governor. No phrase structures are generated and the dependency structures consist of only binary bilexical relations. A CYK-like algorithm is used to construct optimal non-constituent structures that ‘span’ chunks of contiguous words until the ‘span’ covers the whole input.

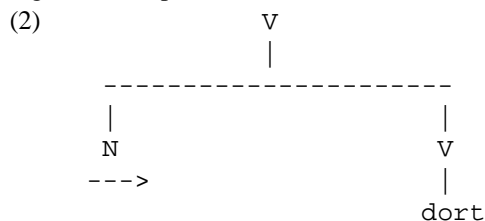
In the experiment reported, a training set of about 40 M of text was taken from a two-gigabyte Chinese newspaper corpus. A lexicon of about 60,000 entries was generated. The performance of the statistical dependency parser was gauged against the annotations in training corpus. For the more stringent criterion of getting both the dominance relation and the functional label correct at the same time, closed and open test averages were 94.7% (95.6% correct in the training corpus) and 74.2% (94.9% correct in the training corpus).

3. Dependency and Lexicalized TAG

The formal properties of this dependency grammar formalism can be compared with those of lexicalised TAG as described in (Abeillé, 1993; Abeillé and Rambow, 2000).

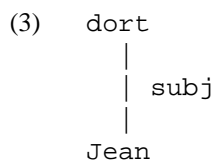
3.1. Initial Trees and Substitution

TAG trees are derived by applying the operations of substitution and adjunction to *initial* trees and *auxiliary* trees respectively. Initial trees like (2) account for the complements in the projection of a subcategorizing word (e.g. *dort* ‘sleeps’).



In this example, the arrow attached to the node N indicates that a similar initial tree for a noun (an N-tree) can replace the node by the substitution operation. To avoid confusion, we avoid the use of the word ‘head’, but *dort* can be safely called the ‘anchor’ of the tree, which forms a part of its lexical property.

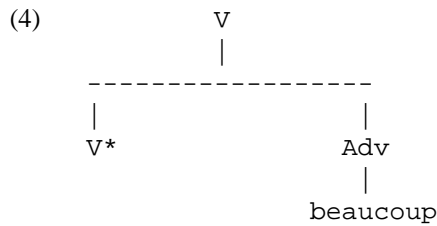
In the dependency grammar formalism, a verb like *dort* will subcategorize for each of its complement dependents.



A word subcategorizes for each of its complement dependents separately. Positional constraints can be added to handle multiple complements.

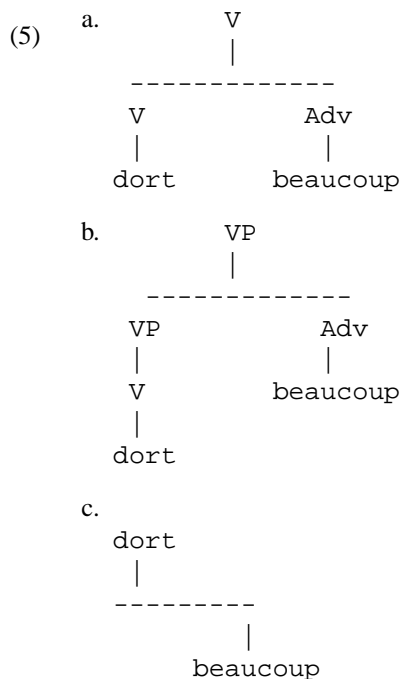
3.2. Auxiliary Trees and Adjunction

In TAG, adjuncts are accounted for by auxiliary trees and the operation of adjunction. For example, the word *beaucoup* ‘a lot’ is the anchor of the auxiliary tree (4):



This auxiliary tree is a lexical property of the anchor *beaucoup*. It can be used to replace a V-tree (a sub-tree) in a syntactic structure. The replaced V-tree is then, used to replace, in turn, the V* node in the auxiliary tree. The *beaucoup* auxiliary tree is ‘adjoined’ to a tree representing the sentence *Jean dort* to obtain *Jean dort beaucoup*.

In TAG, an adjunct is on the same level as or higher than the head word of the phrase in the *derived* syntactic tree as in (5a) or (5b).



TAG is probably correct in letting adverbs (e.g. *beaucoup*) decide that they are to be adjoined to verbs.

In the dependency grammar formalism, a verb (e.g. *dort*) governs an adverb (e.g. *beaucoup*). (See (5c) and the dependency structure for *Jean dort beaucoup* in (1b).) However, there is nothing to prevent a dependent adjunct from determining (as in TAG) what it should be adjoined to.

Where only initial trees and substitutions are involved, it is obvious that TAG derived trees can be pruned into dependency structures. (Dependency relation labels like ‘subject’ can be obtained from lexical information or from the configuration of the tree.) Conversely, dependency structures can also be fleshed out to form TAG derived trees (with minimal structure). Adjunction makes the situation somewhat more complicated as the adjunct has to be placed higher up than the ‘head’ word in the TAG derived tree. This is possible because the substitution-adjunction distinction is obtainable from dependency relation labels (e.g. ‘adjunct’). in the dependency structure.

Similar grammatical information can be stored in the lexicon in either formalism. Additional mechanisms, e.g. feature unification, can also be added on top of the tree backbone in both formalisms.

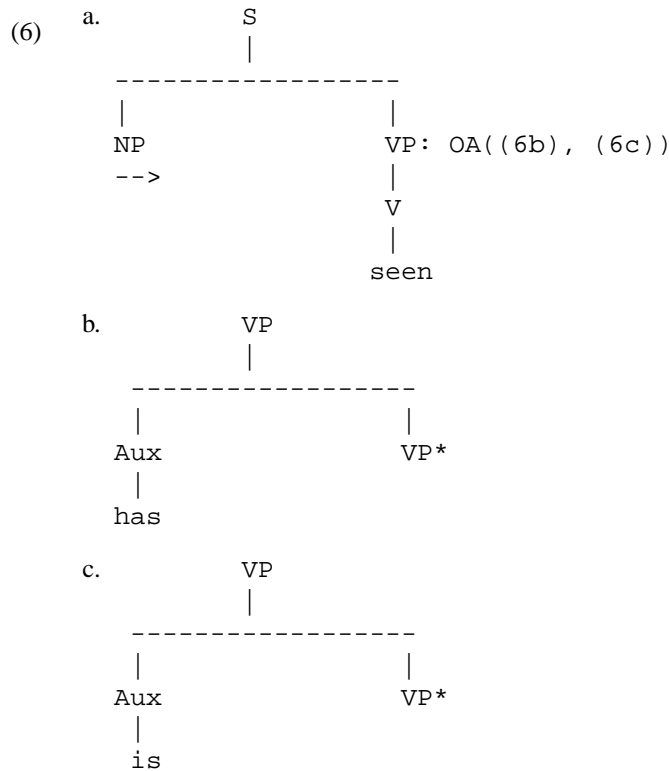
4. Some Complications

The adjuncts (daughters in dependency grammars) that we have looked at are *optional*.

However, with the presence of *adjunction constraints* or *top* and *bottom features* in TAG, adjuncts can either be optional or obligatory.

4.1. Auxiliary Adjunction

One kind of obligatory adjuncts discussed in Abeillé and Rambow (2000) are auxiliaries like *has* and *is* in *has seen* and *is seen*.



The initial tree (6a) associated with *seen* (a past participle form) has a VP node with the *obligatory adjunction constraint* *OA(6b, 6c)* indicating that an adjunction operation must be applied to it using the auxiliary trees (6b) or (6c).

In a dependency grammar, the first decision to be made is whether the governor should be the participle *seen*, or the auxiliary *has* (or *is*). One can very well follow TAG and say that the participle is the governor and the auxiliary is an adjunct daughter. A mechanism can be added to stipulate that, given that the governing verb is in participle form, the adjunct auxiliary is obligatory.

An ‘obligatory’ adjunct may sound weird to those who are accustomed to associating the term ‘adjunct’ with optional dependents, but it should be noted that formally, or mathematically, there is nothing to censor this usage. Anyway, there is obviously a sense in which an auxiliary is not subcategorized for like the ‘complement’ arguments of a predicate.

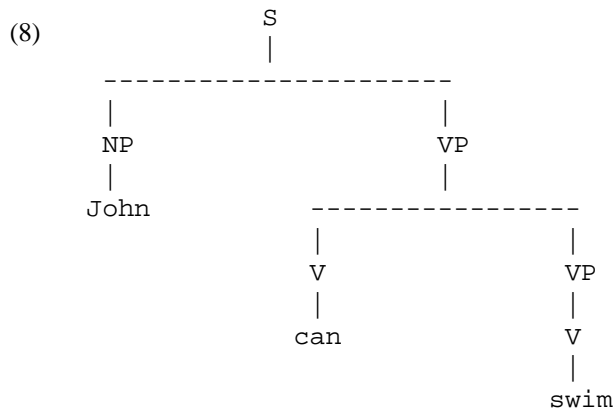
In Chinese, adverbial particles like *le* (perfective marker) are often said to be words rather than morphological affixes. Given this practice, considering these words to be adjuncts as in TAG is not an unreasonable way to account for their occurrence with the governor predicate. It should perhaps be noted, though, that this will also mean that an adjunct can come between the governor predicate and subcategorized complements as in:

- (7) na le dian qian
 take PERF some money
 ‘taken/took some money’

4.2. Modals and Raising Predicates

In Abeillé and Rambow (2000), modals, like *can*, and other raising predicates, like *difficult* and *possible*, are also considered to be obligatory adjuncts.

In TAG, even though modals are adjuncts, they occupy a higher position than the verb predicates in the derived trees.



In dependency grammar, however, the modal will have to be a daughter of the predicate verb.

With other raising predicates, there may be a difficulty. For example, placing *difficult* under *(to) read* in the dependency structure for *difficult to read* will be a remarkable commitment.

It must be said that this is not a problem with TAG. Raising predicate are higher up in derived syntactic tree anyway. This is a problem with dependency grammar only.

5. Concluding Remarks

The formal properties of TAG are well understood (Joshi, Vijay-Shanker and Weir, 1991), and the close relation between TAG and dependency grammar have been known (Rambow and Joshi, 1997)¹. In this paper, in particular, we have noted that the basic TAG mechanisms of substitution and adjunction go well with a projective bilexical dependency grammar approach in general. We have however also noted that coping with the TAG notion of obligatory adjuncts, e.g. as applied to modals and other raising predicates, can be problematic.

To a certain extent, the above observations support the idea of trying to abstract away from particular grammar formalisms when marking the surface syntactic structures of a corpus, for example, as suggested by the annotators of the Chinese PennTreeBank (Xia *et al.*, 2000). It must, however, also be noted that trying to mirror TAG derived trees may be complicated sometimes.

References

- Abeillé, Anne. 1993. *Les nouvelles syntaxes*. Paris: Armand Colin.
- Abeillé, Anne and Owen Rambow. 2000. Tree Adjoining Grammar: An Overview. In Anne Abeillé and Owen Rambow, editors, *Tree Adjoining Grammars: Formalisms, Linguistic Analysis and Processing*. CSLI Publications, Stanford, pages 1–68.
- Collins, Michael. 1996. A New Statistical Parser Based on Bigram Lexical Dependencies. In *Proceedings of 34th Annual Meeting of ACL (ACL'96)*, Santa Cruz.
- Eisner, Jason. 1996. Three New Probabilistic Models for Dependency Parsing: An Exploration. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, pages 340–345, Copenhagen, August.
- Eisner, Jason. 2000. Bilexical Grammars and Their Cubic-time Parsing Algorithms. In Bunt and Nijholt, editors, *Advances in Probabilistic and Other Parsing Technologies*. Kluwer Academic Publishers, Dordrecht, pages 29–62.
- Gaifman, Haim. 1965. Dependency Systems and Phrase-Structure Systems. *Information and Control*, 8:304–337.
- Hays, David. 1964. Dependency Theory: A Formalism and Some Observations. *Language*, 40:511–525.
- Hellwig, Peter. 1986. Dependency Unification Grammar. In *Proceedings of the 11th International Conference on Computational Linguistics (COLING'86)*, pages 195–199.
- Joshi, Aravind, J. Vijay-Shanker and David J. Weir. 1991. The Convergence of Mildly Context-Sensitive Grammar Formalisms. In P. Sells, Shieber S. and T. Warsaw, editors, *Foundational Issues in Natural Language Processing*. MIT Press, Cambridge, Mass., pages 31–81.
- Lai, Tom B.Y. and Changning Huang. 1998. An Approach to Dependency Grammar for Chinese. In Yang Gu, editor, *Studies in Chinese Linguistics*. Linguistic Society of Hong Kong, Hong Kong, pages 143–163.
- Lai, Tom B.Y. and Changning Huang. 2000. Dependency-based Syntactic Analysis of Chinese and Annotation of Parsed Corpus. In *Proceedings of ACL'2000*, pages 255–262, Hong Kong, October.

1. Thanks to an anonymous reviewer for pointing out and providing this reference.

- Lai, Tom B.Y., C.N. Huang, Ming Zhou, J.B. Miao and Tony K.C. Siu. 2001. Span-based Statistical Dependency Parsing of Chinese. In *Proceedings of NLPRS'2001*, pages 667–684, Tokyo, November.
- Rambow, Owen and Aravind Joshi. 1997. A Formal Look at Dependency Grammars and Phrase-Structure Grammars, with Special Consideration of Word-Order Phenomena. In Leo Wanner, editor, *Recent Trends in Meaning-Text Theory*. John Benjamins, Amsterdam and Philadelphia, pages 167–190.
- Robinson, Jane. 1970. Dependency Structures and Transformation Rules. *Language*, 46:259–285.
- Tesnière, Lucien. 1959. *Elements de syntaxe structurale*. Paris: Klincksieck.
- Xia, Fei, Martha Palmer, Nianwen Xue, Mary Ellen Okurowski, John Kovarik, Fu-Dong Chiou, Shizhe Huang, Tony Kroch and Mitch Marcus. 2000. Developing Guidelines and Ensuring Consistency for Chinese Text Annotation. In *LREC-2000*, Athens.