

FREE WORD ORDER IN A CONSTRAINT-BASED IMPLEMENTATION OF DEPENDENCY GRAMMAR

Tom B.Y. Lai

Department of Chinese, Translation and Linguistics,
City University of Hong Kong
Email: cttomlai@cityu.edu.hk

Huang Changning

Department of Computer Science and Technology,
Tsinghua University, Beijing
Email: hcn@mail.tsinghua.edu.cn

ABSTRACT

Parsing of sentences based on Dependency Grammar is emulated with a constraint- and unification-based mechanism that preserves single-headedness and projectivity in syntactic dependency. Working on Chinese, and referring to English at times, the authors have treated subcategorization properties of verbs on the assumptions that relative positions of obligatory complements of a verb are fixed, and that optional verbal adjuncts, while relatively free in their choice of positions in the sentence, are constrained to be placed farther away from the governing verb than any complements. A pair of subcategorization lists residing in the governing verb have been used to capture these properties. However when one tries to extend the model to languages with relatively free word order like Japanese, for which the assumptions mentioned above are not valid, straight-forward list-manipulations on subcategorization lists will not be adequate. This paper discusses how additional subcategorization handling mechanisms can be introduced to deal with verbal complements scrambling and intervening adjuncts, while retaining the mechanisms to capture default word order.

1. INTRODUCTION

Computational linguists in China [9][26][27] have been studying and experimenting with Chinese sentence parsing based on the tradition of Gaifman, Hays and Robinson [4][7][21] in the development of Dependency Grammar,[25] which is different from other traditions of the grammar formalism [10][24][19][6] in that *single-headedness* and *projectivity* are required for dependency relations holding between *governor-dependent* pairs of words in a sentence. Using different algorithms for the parsing process, they differ in whether syntactic categories are considered in lieu of the actual words and in whether and how word order is taken into consideration, but they all observe the requirements of *single-headedness* and *projectivity*. They also label dependency relations with functional labels like *subject* and *object* as is done in practically all schools of Dependency Grammar.

Unification [22] has been used in Dependency Grammar implementation and theory (e.g. [8][2][18]). The authors have distinguished between a *single-headed* and *projective* dependency (constituent) structure and a much less constrained functional structure.[14][15][16][17] Dependency rules [7] are annotated with functional annotation,[1] and a unification-based parser [22][5] has been adapted to produce dependency and functional structures for Chinese (and English) sentences. They have discussed how syntactic phenomena like control are dealt with. They make a distinction between obligatory complements prescribed for by the syntactic properties of the governing verb and optional adjuncts that are taken care of by non-lexical rules in the syntax of the language. A pair of subcategorization lists residing in the lexical entry of the governing verb are used to take care of obligatory verbal complements, making use of constraints in Chinese (and English) that the subject is to the left of the verb while the other verbal complements are on the other side, and that optional adjuncts must not come between the verb and the non-subject complements.

Though basically respected in Chinese (and English), these constraints are obviously too restrictive. In Japanese, for example, the subject, obligatory verbal complements and optional verbal adjuncts are all on the same side of the verb, complements can be “scrambled”, and adjuncts often come between complements and the verb.

In this paper, the authors’ mechanisms for dealing with verbal complements and adjuncts mentioned above [17] are modified and extended to take care of free word order. Syntactic properties of the governing verbs are encoded in their dictionary entries, and, free complement and adjunct word order are taken care of by making modifications to the subcategorization lists and introducing special processing mechanism, while inherent order of elements in the subcategorization list is still useful in capturing default or preferred word order.

2. DEPENDENCY RULES, FUNCTIONAL ANNOTATION AND SUBCATEGORIZATION

2.1 Single-Headed and Projective Dependency Grammar

In the authors’ Dependency Grammar formalism, words contract governor-dependent (head-modifier) relationships as shown below for the Chinese sentence *na ren zai gongyuan li* (from [26]):

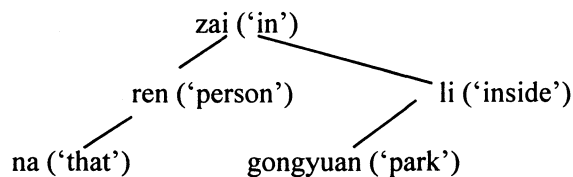


Fig. 1

The dependency structure in Fig. 1, which has been drawn so that the left-to-right word order of the sentence is reflected, can also be represented as in Fig. 2:

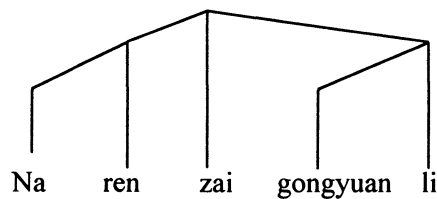


Fig. 2

These structures are governed by Robinson’s four axioms of well-formedness:[21]

- (a) one and only one element is independent;
- (b) all others depend directly on some element;
- (c) no element depends directly on more than one other;
- (d) 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.

It can be seen that the third axiom is a requirement of *single-headedness*. The fourth axiom is a requirement of *projectivity* in the sense that structures like Fig. 3, in which branches cross one another, are not allowed.

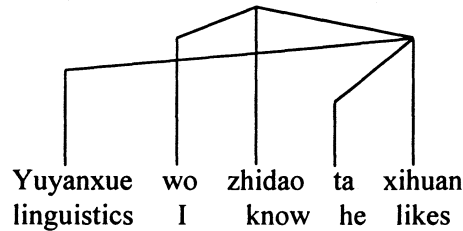


Fig.3

Dependency structures with these properties can also be accounted for by Hays' [7] *dependency rules*:

- (a) $X(A,B,C, \dots, H, *, Y, \dots, Z)$
- (b) $X(*)$
- (c) $*(X)$

Here, (a) states that the governing *auxiliary alphabet* X has dependents A, B, C, ..., H, Y, ..., Z and that X itself (the governor) is situated between H and Y. (b) says that the *terminal alphabet* X occurs without any dependents. In (c), X occurs without any governor, i.e. it is the *main* or *central* element.

Gaifman [4] has established that a Dependency Grammar obtained in this way, which is *single-headed* and *projective*, is equivalent to a phrase structure grammar in the sense that:

- (a) they have the same terminal alphabet;
- (b) for every string over that alphabet, every structure attributed by either grammar corresponds to a structure attributed by the other.

2.2 Dependency Rules and Functional Annotation

Not reflected in the formal representations reviewed above, grammatical functions like *subject* and *object* are generally used to label dependency links. Applying the idea of *functional annotation*, [1] to Hays' rules, the authors have proposed *annotated dependency rules* like the following: [14]

$$X(A(\text{fa}), B(\text{fb}), \dots, *, \dots, Z(\text{fz}))$$

For example, the following rules account for the transitive verbs:

- (a) $*(\text{TV})$
- (b) $\text{TV}(\text{N}(\text{subj}), *, \text{N}(\text{obj}))$
- (c) $\text{N}(*)$

The authors [16] have implemented a Dependency Grammar parser using PATR. [22][5] In this implementation, the Chinese sentence *Zhang San (name) kanjian ('saw') Li Si (name)* is accounted for by the following PATR rule and appropriate lexical entries: (For details, see [16].)

$$\begin{array}{lll} \text{TV} \rightarrow \text{N1} \text{ TV} \text{ N2} & \% \text{ only details of immediate significance shown} \\ \text{TV.ds.subj} = \text{N1.ds} & \\ \text{TV.ds.obj} = \text{N2.ds} & \end{array}$$

Fig. 4

Besides a grammatical function structure, the following dependency structure is produced:

[tv, [[n, [zhangsan]], [tv, [kanjian]], [n, [lisi]]]]

After applying a pruning operation on the two *tv*'s (one immediately dominating the other), a dependency structure is obtained. It is in this sense that the parser adapted from PATR is conceived as a Dependency Grammar parser.

More complicated sentence, e.g. *Zhang San (name) xiang ('want') da ('hit') Li Si (name)*, which involves a subject-control verb, are dealt with by constraints defined in terms of dependency links (functional features). (For details, see [16].) By introducing a level of grammatical function (dependency type) to accommodate the offending complications, the authors [15][16] preserve *single-headedness* and *projectivity* in syntactic dependency. Morphological and semantic constraints can also be dealt with on additional grammatical levels.

2.3 Complements and Adjuncts in Chinese and English

While some Dependency Grammars (e.g. [24]) require that the rules of grammar should not be formally distinct from subcategorization facts, the authors' approach recognizes that there are syntactic phenomena that are not triggered by lexical properties. Verbal adjuncts are an example.

To deal with adjuncts, the authors use, in PATR notations, rules like: [17]

```
X → X Y                % similar rules for left adjuncts
  Y.fun = adjunct
```

Fig. 5

In Chinese and English, adjuncts generally do not come between a head word and its 'unmoved' non-subject complements. Exploiting this property, lexical entries are endowed with separate subcategorization lists [20] for the subject and for the other obligatory complements as follows

```
gei ('give')                % immediately significant features only
  subcat.left = [n(subj)]
  subcat.right = [n(iobj), n(obj)]
```

Fig. 6

Rules like the following (necessary details only, for brevity) take care of complements subcategorized for by the head word:

```
V → V X
  {Z = pop(V.subcat.right)}    % Z = cat(fun) % fails if V.subcat.right is empty
  X.cat = functor of Z        % functor of Z = cat
  X.fun = argument of Z      % argument of Z = fun
```

Fig. 7

A similar rule $V \rightarrow X V$ using the list $V.subcat.left$ accounts for the subject. Adjuncts are kept from getting in between a head word and its unmoved non-subject complements by adding to the adjunct rules in Fig. 5 the following constraint:

```
X.subcat.right = empty list
```


In general, subjects and other complements of a Japanese verb can be “scrambled” without making the sentence ungrammatical. This requires that the strictly order-respecting *pop* action for retrieving elements in the subcategorization list be modified.

Another difficulty with Japanese is that optional adjuncts often intervene between the verb and the complements.

```

John   ga  sono   hon o  ima yonde  iru.   ([3]: 102)   (7)
person subj that  book obj now reading
‘John is now reading that book’
Watashi wa Asakusa kara Soka e junkyu de ikimashita. ([24]: 124) (8)
I       topic place from place to semi-express by went
‘I went from Asakusa to Soka by the semi-express train’

```

4 PARSING MECHANISMS FOR FREE COMPLEMENT AND ADJUNCT ORDER

Modifications to take care of argument scrambling and adjunct intervening are rather straight forward.

First, the modification to allow argument scrambling is to replace the *pop* action in Fig. 7 with a *find_and_get* action as in Fig. 10:

```

V → X V           % re. word order in Japanese
{Z = find_and_get(V.subcat.compl)} % Z = cat(fun)
% fails if V.subcat.compl = empty list
X.cat = functor of Z
X.fun = argument of Z

```

Fig. 8

The *find_and_get* action searches a list for an element, returning it and removing it on success. In Japanese, the subject and other complements are on the same side of the governing verb. The distinction between the subject and other complements can be accounted for by a similar rule using the list *V.subcat.subj*.

To allow intervention of adjuncts between the verb and the complements, one needs only to make sure that the adjunct rules are not constrained by any requirement that subcategorization lists be empty:

```

X → Y X
% X.subcat.compl = empty list   % X.subcat.subj = empty list
% the above two constraints must not apply
Y.fun = adjunct

```

Fig. 9

These modifications do not change the essence of the authors’ system. With or without these modifications, the system has the following characteristics:

- (a) A set of annotated dependency rules, with parallel generation of syntactic dependency structures and ‘functional’ structures ensuring that complicated grammatical constraints can be satisfied within a single-headed and projective dependency framework;
- (b) Subcategorization information is stored in the lexical entries;

- (c) Adjunction is not triggered by lexical properties of the governing head;

It may seem ridiculous to have a list model to capture fixed complement order in some languages and then undo the inherent ordering of list elements when extending the model to languages with free complement order (scrambling). However, to the best of the authors' knowledge, all languages with free word order (e.g. Japanese, Mongolian, Tagalog) do indeed have a *preferred* word order. Preferred or default word order can be captured by the subcategorization list quite conveniently. For example, in the implementation described in this section, preferred word order can be kept alive introducing *alternatives*:

```
{ Z = pop(V.subcat.subj);           % this alternative will be tried first
  Z = find_and_get(V.subcat.subj)}
```

to the list operations in Fig. 8, provided care is taken that the *pop* action removes the list element if, and only if, it succeeds. This very crude implementation brings about a proliferation of backtracking, but may recognize correct sentences in the default word order efficiently under the right conditions. In sophisticated systems, it may be possible to vary the role played by the *find_and_get* action depending of the degree of word order flexibility (completely turned off in the case of fixed word-order languages).

The mechanism allowing and not allowing intervention of adjuncts between complements and the subcategorizing verb is of a different nature. It works in the other direction in the sense that if a constraint is not included, adjunct intervention will be allowed.. Suppose that the adjunct rules become:

```
X → Y X
((X.subcat.compl. = empty list, X.subcat.subj = empty list); true)
Y.fun = adjunct
```

Fig. 10

The first alternative applies the constraint (as in the case of Chinese). If this fails, the *true* alternative, which allows the adjunct to intervene between complement and head, will *always* be satisfied. The introduction of mechanisms for variable constraint strength (in the logic programming paradigm of PATR is considerable.

5 CONCLUSION

The Dependency Grammar formalism described in this paper features annotated dependency rules; parallel generation of a single-headed and projective syntactic dependency structure and a much less constrained functional structure; lexically triggered syntactic phenomena like subcategorization working in conjunction with non-lexical syntactic rules like adjunction and topicalization; a list-based model with inherent word order for encoding subcategorization information; a free adjunct positioning mechanism unless it is constrained for non-intervention between complement and subcategorizing head. Implemented using PATR (in PROLOG) for convenience, this formalism, which was first developed for a fixed-word-order language, seems also to have potentials for use with languages with free complement and adjunct order.

6 ACKNOWLEDGMENTS

This research is supported by Grant 7000888 of City University of Hong Kong and the National Science Foundation of China.

7 REFERENCES

- [1] R.M. Kaplan and J. Bresnan. "Lexical-Functional Grammar: A Formal System for Grammatical Representation," In J.W. Bresnan. (ed.) *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, U.S.A., pp. 173-281, 1982.
- [2] M.A. Covington. "Parsing Discontinuous Constituents in Dependency Grammar," *Computational Linguistics*, 16/4, pp. 234-236, 1990.
- [3] N. Fukui. *Theory of Projection in Syntax*. Stanford: CSLI, 157p, 1995.
- [4] H. Gaifman. (1965) "Dependency Systems and Phrase-Structure Systems," *Information and Control*, 8, pp. 304-337, 1965.
- [5] G. Gazdar and C. Mellish. *Natural Language Processing in Prolog*. Addison Wesley, Wokingham, 504p, 1989.
- [6] E. Hajicova. "Free Word Order Described Without Unnecessary Complexity". *Theoretical Linguistics*, 17, pp. 99-106, 1991.
- [7] D.G. Hays. "Dependency Theory: A Formalism and Some Observations," *Language*, 40, pp. 511-525, 1964.
- [8] P. Hellwig. "Dependency Unification Grammar," COLING'86, pp. 195-199, 1986.
- [9] C.N. Huang, C.F. Yuan and S.M. Pan. "Yuliaoku, Zhishi Huoqu He Jufa Fenxi (Corpora, Knowledge Acquisition and Syntactic Parsing)," *Journal of Chinese Information Processing*, 6/3, pp. 1-6, 1992.
- [10] R. Hudson. *Word Grammar*. Blackwell, Oxford, 267p, 1984.
- [11] R. Hudson. *English Word Grammar*. Blackwell, Oxford, 445p, 1990.
- [12] R. Hudson R. "Discontinuous Phrases in Dependency Grammar," *University College London Working Papers in Linguistics*, 6, pp. 89-124, 1994.
- [13] R. Hudson. (1995) "Dependency Counts," In E. Hajicova (ed.) *Functional Description of Language*, Faculty of Mathematics and Physics, Charles University, Prague, pp. 85-115, 1995.
- [14] B.Y. Lai and C.N. Huang. "Dependency Grammar and the Parsing of Chinese Sentences," 1994 Kyoto Conference (Joint ACLIC8 and PACFoCoL2), pp. 63-71, 10-11 Aug. 1994.
- [15] B.Y.T. Lai and C.N. Huang C.N. "Single-Headedness and Projectivity for Syntactic Dependency," *Linguistics Association of Great Britain Spring Conference*, University of New Castle , 10-12 August, 1995.
- [16] T.B.Y. Lai and C.N. Huang. "An Approach to Dependency Grammar for Chinese," In Y. Gu (ed.) *Studies in Chinese Linguistics*, Linguistic Society of Hong Kong, Hong Kong, pp. 143-163, 1998.
- [17] T.B.Y. Lai and C.N. Huang. "Complements and Adjuncts in Dependency Grammar Parsing Emulated by a Constrained Context-Free Grammar," COLING-ACL'98 Workshop on Processing of Dependency-Based Grammars, Montreal, pp. 102-108, 1998.
- [18] D. Maxwell. *Unification Dependency Grammar*, manuscript.
- [19] I.A. Melcuk I.A. *Dependency Syntax: Theory and Practice*. State University of New York Press, Albany, 428p, 1988.
- [20] C. Pollard and I. Sag. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago, 440p, 1994.
- [21] J.J. Robinson "Dependency Structures and Transformation Rules," *Language*, 46, pp. 259-285, 1970.
- [22] S.M. Shieber. *An Introduction to Unification-Based Approach to Grammar*. Chicago University Press, Chicago, 105p, 1986.
- [23] Y. Shimokawa. *Gendai Nihongo Kobunpo: Okubo Bunpo No Keisho To Hatten (Modern Japanese Syntax: Continuation and Development of Okubo's Grammar)*. Sanseido, Tokyo, 347p, 1993.
- [x] S. Starosta S. *The Case for Lexicase*. Pinter, London, 273p, 1988.
- [25] L. Tesnière L. *Elements de syntaxe structurale*, Klincksieck, Paris, 1959.
- [26] C.F. Yuan and C.N. Huang. "Knowledge Acquisition and Chinese Parsing Based on Corpus," COLING'92, Nantes, France, pp. 13000-13004, 1992.
- [27] M. Zhou and C.N. Huang. *An Efficient Syntactic Tagging Tool for Corpora*. COLING'94, Kyoto, pp.949-955, 1994.