

# Dependency-based Syntactic Analysis of Chinese and Annotation of Parsed Corpus

**Tom B.Y. LAI**

City University of Hong Kong  
Tat Chee Ave., Hong Kong  
Tsinghua University, Beijing  
Beijing 100084  
cttomlai@cityu.edu.hk

**HUANG Changning**

Microsoft Research, China  
5F, Beijing Sigma Center  
49 Zhichun Road, Beijing 100080  
cnhuang@microsoft.com

## Abstract

Following a dependency-based framework that admits no intermediate phrasal nodes and allows no crossing of syntactic dependency links, we discuss how Chinese sentences are analysed and annotated using an SGML-based scheme. Issues related to tolerance of errors at various levels of analysis and compatibility with other syntactic frameworks are addressed.

## 1 Introduction

We annotate the syntactic dependency between lexical units in a linguistic expression according to an approach to Dependency Grammar (Tesnière, 1959) that can be traced back to (Gaifman, 1965; Hays, 1964; Robinson, 1970). We require syntactic dependency to be projective, and capture non-projective syntactic phenomena in language with constraints defined with reference to this projective skeleton (Lai and Huang, 1998b; Lai and Huang, 1998a; Lai and Huang, 1999a; Lai and Huang, 1999b). This is different from approaches that allow multiple heads and crossing dependency links (Hudson, 1984; Mel'čuk, 1988; Starosta, 1988; Hajičova, 1991). We also require the relationship between a governor and all its dependents to be immediate. There are no intermediate phrasal nodes as in some other projective-dependency-based parsing approaches (Hellwig, 1986; Covington, 1990; Courtin and Genthial, 1998; Bourdon et al., 1998; Nagao, 1993). A similar approach has been adopted in some recent works

on Chinese (Huang et al., 1992; Yuan and Huang, 1992; Zhou and Huang, 1994).

As a start, we manually give parse structure annotation to a small Chinese legal text corpus. The text is “segmented” and tagged, and morphological analysis is then carried out to confirm and adjust word boundaries. Syntactic units larger than the word are annotated using an SGML-based scheme. We have also found it useful to identify units smaller than the word like *characters* and *dictionary entries*.

In the sections that follow, we give a short description of the underlying dependency-based grammar formalism, a description of the annotation scheme. We also discuss some morphological and syntactic complications, and their implications for tolerance of errors and compatibility with other grammar formalisms.

## 2 Projective dependency syntax

We follow Robinson (1970) in requiring syntactic dependency structures to observe the constraints of *single-headedness* and *projectivity*.

Robinson requires that all words (except for the head of the whole expression) should depend on exactly one other word in the expression. She also requires that no dependency links should cross any other. For example, the projective syntactic dependency structure for the Chinese sentence *ta xiang xiao* (‘he wants/wanted to laugh’) is as in Figure 1. There is no explicit syntactic link between *ta* and *xiao*. The fact that *ta* is the subject of *xiao* is accounted for by the “control” specification in the lexical entry of the word *xi-*

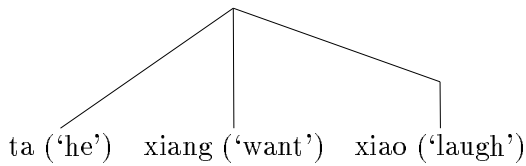


Figure 1: Projective dependency skeleton

*ang* that its subject is the subject of its predicate complement. Grammatical constraints like these propagate via the nodes and arcs of a projective syntactic dependency skeleton. Constraints that account for movement-like phenomena work similarly.

We use dependency rules (Hays, 1964) to generate dependency structures. We make the dependency rules (1) binary-branching.

- (1) a.  $H \rightarrow X H$

Subject to constraints  
in lexical entry of H

- b.  $H \rightarrow H X$

Subject to constraints  
in lexical entry of H

Repeated application of these two dependency rules under the constraints of the subcategorization properties of the governing word and global constraints of the language accounts for dependent elements in its “domain”. The actual order in which the dependents are arranged depends on how the subcategorization information in the lexical entry of the governor is organized, and on how subcategorization information is retrieved for use in the dependency rules.

In the binary rules (1), the *H* symbols on the two sides of the rule are not only of the same *type*, as in phrase-structure rewrite rules, but identical as *tokens*. A linguistic expression is thus indistinguishable from the subcategorizing head word. A head word and its dependents form a flat structure without intermediate phrasal nodes.

### 3 Dependency-based annotation

#### 3.1 Pilot study with a small text corpus

To study how we will be able to annotate corpora according to our dependency-based approach in a way that the corpora will also be useful to other researchers, we begin with manually annotating a small corpus. It is also hoped that this piloting effort will throw light on how we may make use of large annotated corpora produced by other people.

We use a small corpus of Chinese text segmented and tagged using a bigram-based segmentation-tagging tool (Lai et al., 1992; Lai et al., 1998). The corpus is two “chapters” of a statute in Hong Kong containing 4797 word tokens.

#### 3.2 The annotation scheme

The annotation scheme is based on SGML (SGML, 1986). The largest tag unit is the *parsing unit* (`<pu>`) that marks off chunks of text that are ready to be processed by a syntactic parser. It has a *pi* attribute to reflect its position in the text. For example:

```
<pu pi=3>"7" "." "1"
  <!--'7.1'--></pu>
<pu pi=4>"xing2" "shi4"
  <!--'form'--></pu>
<pu pi=5>"7" "." "1" "." "1"
  <!--'7.1.1'--></pu>
<pu pi=6>"he2" "yue4" "yin1"
  "xing2" "shi4" "bu4" "he2"
  "gui1" "ge2" "ke3" "neng2"
  "wu2" "xiao4" "huo4" "bu4"
  "neng2" "li3" "xing2" ".."
  <!--'contracts rendered
  invalid or becoming impossible
  to carry out because their
  forms do not conform to
  specifications'--></pu>
```

In this example, the parsing units are a section number in Arabic numerals, a section caption, a subsection number, and a one-sentence text. Parsing units may be sentences (e.g. `<pu pi=6>`), phrases (e.g. `<pu pi=4>`), or non-phrasal units like `<pu pi=3>`

and  $\langle pu\ pi=5 \rangle$ . They are the units from which to build larger discourse units (which is beyond the scope of this paper).

Words form the units of syntactic analysis. For example:

```
<pu pi=6>
<wu wi=1 gv=7 fn=sub ct=nc
  sm="contract">"he2" "yue4"
<wu wi=7 gv=0 ct=aux sm="may">
  "ke3" "neng2"
<wu wi=8 gv=7 fn=axo ct=aj
  sm="invalid">"wu2" "xiao4"
<wu wi=13 gv=7 fn=pt ct="..">.."
</pu>
```

The  $\langle wu \rangle$  elements have a  $wi$  (for *word index*) attribute that marks their positions within the  $\langle pu \rangle$  unit. (Some words in the example are omitted for the sake of clarity.) Dependency hierarchy is marked with the help of the  $gv$  (for *governor*) attribute, which records the  $wi$  value of the governor of a word. A value of 0 for  $gv$  shows that the word is the head element of the  $\langle pu \rangle$ . The syntactic category of the word is given in the  $ct$  (for *category*) attribute, and its “grammatical” relation to its immediate governor is the value of the  $fn$  (for *function*) attribute. The semantic glosses of the words are given in their  $sm$  attributes.

Words are formed from *dictionary units* ( $\langle du \rangle$ ), forms that are listed in Chinese dictionaries.

```
<wu wi=1 ct=nc sm="contract">
  <du tg=ncd><cu>"he2"<cu>"yue4"
```

Dictionary units are made up of one or more characters ( $\langle cu \rangle$ ). Words and dictionary units are considered different kinds of entities because of morphological complications to be discussed later in this paper. The relation of the  $tg$  (for *tag*) attributes of dictionary units ( $du$ ) to the  $ct$  attributes of the words will be explained later.

The characters ( $\langle cu \rangle$ ) are the terminal text elements in our annotation scheme. They are multiple-byte Chinese (*Han*) characters.

In this paper, they are written in the phonetic *pinyin* script for the ease of rendering and reading.

In our annotation scheme, the tags  $\langle wu \rangle$  and  $\langle cu \rangle$  correspond to the  $\langle w \rangle$  and  $\langle c \rangle$  tags for *linguistic segmentation elements* in CES (Ide et al., 1996). The *morphological unit* (to be introduced and discussed in the next section) is, however, not of the same nature as the  $\langle m \rangle$  tag in CES. We do not have tags corresponding to  $\langle cl \rangle$  and  $\langle phr \rangle$  in CES.

Thus, parsing units ( $\langle pu \rangle$ ) in our annotation scheme are flat ensembles of words ( $\langle wu \rangle$ ). The governor ( $\langle gv \rangle$ ) attributes of the words reflect the dependency hierarchy. Words are made up of dictionary units ( $\langle du \rangle$ ), which are in turn made up of characters ( $\langle cu \rangle$ ). The basic text elements are the characters. Morphological complications are discussed in the next section.

## 4 The sub-word units — characters, dictionary entries and morphemes

### 4.1 Characters and dictionary entries

The motivation for the use of the dictionary unit tag is explained with the help of the following example.

```
<pu pi=3>
<wu wi=1 gv=0 ct=mx sm="7.1">
  <du tg=mx><cu>"7"
  <du tg="."><cu> "."
  <du tg=mx><cu>"1"
</pu>
```

Our segmentation-tagging module produces three separate “tokens”. We adjust the word boundaries and group the three one-character tokens together into one single word ( $\langle wu \rangle$ ), which is a kind or numerical label.

The segmentation-tagging module segments the text into tokens according to criteria popular among researchers in China Mainland. The three characters 7, “.” and 1 are listed as entries in our dictionary, and they combine transparently to form the word *7.1*, which is, with good reasons, not listed in dictionary. We group the three characters into

a word, but we also retain the information that this word is composed from the three one-character dictionary entries. The dictionary tags assigned to these three tokens are retained in the *tg* attribute of (*<du>*).

Researchers may not agree on how to “segment” an expression like this example *<pu>*. What we consider to be one word may be three words to some people. Because of this, we have taken care that the original characters, as well as the order in which they occur, are retained in the marked up text. This is in accordance with good text encoding practice. Recovery from error and re-analysis by other researchers will also be made easier.

## 4.2 Derivational and inflectional morphology

Although Chinese is often said to be a language without (much) morphology, there are morphological issues that cause complications with the projective-based dependency annotation.

Consider the following two parse units from our corpus:

```
<pu pi=1>
<mu mi=1 wu=1><du tg=hm><cu>"di4"
<wu wi=1 gv=2 fn=nm ct=mx
  sm="seventh"><du tg=mx ><cu>"qi1"
<wu wi=2 gv=0 ct=ncl
  sm="chapter">
  <du tg=cnb><cu>"zhang1"
</pu>
<pu pi=2>
<wu wi=1 gv=0 fn=sub ct=nc mh="de"
  sm="contract">
  <du tg=ncd><cu>"he2"<cu>"yue4"
<mu mi=1 wu=1><du tg=ed><cu>"de"
<wu wi=2 gv=0 fn=sub ct=na
  sm="form">
  <du tg=nad><cu>"xing2"<cu>"shi4"
</pu>
```

In the first parsing unit (*pi* = 1), our segmentation and tagging module outputs *di4* and *qi1* as separate tokens. In the second parsing unit (*pi* = 2), *he2*, *yue4* and *de* are also separate tokens. We consider *di4qi1* (‘seventh’) and *he2yue4de* (‘contract’+GEN)

to be words, but we mark off *di4* and *de* as morphological units (*<mu>*). As morphological units are not words, they do not have *wi* attributes. Instead, the stems that they are attached to are recorded in their *wu* attributes. As they are themselves listed in the dictionary (a common practice in Chinese), they are marked as *<du>* and their *tg* attributes are retained. This is a conscious effort to retain information in the original text.

In the first example, *di4qi1* (‘seventh’) is a *derived* word. The stem *<du>* *qi1* (‘seven’) bears the *<wu>*. The attributes (e.g. category and semantic gloss) of the word unit are those of the derived word.

In the second example, *he2yue4de* (‘contract’+GEN) is an *inflected* form of the word *he2yue4*. The stem carries the word unit, which is marked with an *mh* attribute (for *morphology*). When the inflected word is accessed for syntactic analysis, the affix does not have to be accessed because the required information is already recorded in this attribute.

Further complications with morphological analysis are illustrated by the following examples from a (stylistically more balanced) newspaper corpus we are working on:

```
<pu>
<wu wi=1 gv=0 ct=vn mh="perf"
  sm="have meal">
  <du mu="2" tg=vnm><cu>"chi1"
<mu mi=1 wu=1><du tg=el><cu>"le"
<mu mi=2 wu=1><du tg=ncm><cu>"fan4"
</pu>
<pu>
<wu wi=1 gv=0 ct=adj mh="dup, de"
  sm="happy">
  <du mu="2" tg=ad><cu>"gao1"
<mu mi=1 wu=1><cu>"gao1"
<mu mi=2 wu=1><cu>"xing4"
<mu mi=3 wu=1><cu>"xing4"
<mu mi=4 wu=1><du tg=ed><cu>"de"
</pu>
```

In the first example, an infix *le* is inserted between the two characters of the word *chi1fan4*. As is common in linguistic and computational research on Chinese, our seg-

mentation module “segments” the text rather than “lemmatizes”. It outputs three one-character tokens, which, in Chinese, are all valid dictionary entries. With these tokens as input, we lemmatize, but we take care to make sure that the original output of the segmentation-tagging tool is recoverable. The word unit is accessed through the first non-affix character, and its syntactic and semantic attributes (*ct* and *sm*) have values corresponding to the whole word, not those corresponding to the character.

The second example poses even more problems for the simple markup method in Section 4.1. The word is made up of the single two-character dictionary entry *gao1xing4*. However, the morphological process of reduplication has been applied, and each of the two characters is repeated to give a four-character word. As the segmentation-tagging module does not lemmatize and does not meddle with the order in which the characters occur, its output is (somewhat erroneously) the four-token sequence of *gao1 gao1 xing4 xing4*. These characters are all valid dictionary entries in Chinese themselves, but they do not combine transparently to form the word *gao1gao1xing4xing4*, which is a derived word. In our annotation, only the first *<du>* bears the word unit tag. When the duplicated word is accessed for syntactic analysis, its *<mh>* attribute is available to give the necessary grammatical information. The three trailing *<mu>* are neglected. As shown in the example, when we deal with this kind of morphology, we take care to ensure that the original output of the segmentation tool is preserved.

The second example also contains a morphological affix *de*. It may be noted that the *mh* value of the *<du>* that carries the word unit tag is now a list (in the form of a quoted character string to conform to SGML). This serves to illustrate how multiple affixes attached to a stem are dealt with.

### 4.3 Portability implications

The lack of general agreement among theoretical and computational linguists about how

morphology is treated in Chinese has made it difficult for linguistically annotated resources to be shared. We must adopt a particular approach in our own work, but by preserving the morphemes and characters in their original positions, it is hoped that our annotations will be useful to other researchers.

## 5 Complications with Syntactic Bracketing

### 5.1 Different parse structures according to different approaches

A more serious portability concern of syntactic annotation is the fact that different researchers may prefer to use different grammar formalisms and syntactic theories in their work. Efforts like the PENN Treebank (Marcus et al., 1994) have tried to minimize the differences by adopting a basic bracketing scheme, but the problem is that different approaches may require different ways of bracketing for the same linguistic expression. Consider the following example from our corpus:

```
<pu pi=8>
<wu wi=1 gv=4 fn=mks ct=cnj
  sm="because"><du tg=jom><cu>"yin1"
<wu wi=2 gv=4 fn=sub ct=na
  sm="form">
  <du tg=nad><cu>"xing2"<cu>"shi4"
<wu wi=3 gv=4 fn=neg ct=adv
  sm="neg"><du tg=bu><cu>"bu4"
<wu wi=4 gv=8 fn=ajt ct=vt
  sm="conform"><du tg=vnm><cu>"he2"
<wu wi=5 gv=4 fn=obj ct=na
  sm="stipulation">
  <du tg=nad><cu>"gui1"<cu>"ge2"
<wu wi=8 gv=11 ...>...
<wu wi=17 gv=13 fn=pt ct="..">
  <du tg=".."><cu>".."
</pu>
```

The first word (*wi* = 1) *yin1* (‘because’) is marked as a subordinating conjunction (*ct* = *cnj*). In our annotation, it is governed by the head word (*wi* = 4) of the embedded sentence (*wi* = 1 to *wi* = 5), *he2* (‘conform/satisfy’), for which it functions somewhat like a *marker* in HPSG (Pollard and Sag, 1994). In HPSG,

this is a preposition (that does not have a noun phrase complement), but as it is nevertheless often considered to be a daughter of the (embedded) sentence, there is no serious parse structure disagreement between HPSG and our annotation. However, if one follows more popular approaches like the Chomskyan school, one may prefer to reverse the direction of the dependency — the subordinating conjunction will be higher up in the dependency hierarchy than the predicate verb in the embedded sentence.

As there is no general consensus about what parse structure to give to a linguistic expression, it will be in vain for us to attempt to find parse structures that are generally accepted. What we can do, and have done, is to label arcs of our parse structures with the dependency relation names, thus leaving the hope alive that our parse structures may be convertible for use with other approaches.

## 5.2 Different ways to realize arguments

The usefulness of having the names of dependency relations in the annotation may be further illustrated by the different ways to analyse auxiliaries and modals. Consider the following example (unnecessary details omitted):

```
<pu>
<wu wi=1 gv=0 sm="can">"neng2"
<wu wi=2 gv=1 sm="carry out">
  "li3" "xing2"
<wu wi3 gv=2 sm="contract">
  "he2" "yue4"
</pu>
```

In our annotation, the auxiliary *neng2* (‘can’) governs the verb *li3xing2* (‘carry out’), which governs its own argument daughter *he2yue4* (‘contract’).

However, in recent developments of HPSG motivated by the study of German (Nerbonne et al., 1994) and the Romance languages (Balari and Dini, 1998), the auxiliary verb is considered to pull up the arguments of the lexical verb into its own domain by means of a process known as *argument extraction* or

*composition*. Under such an analysis, the syntactic dependency structure will be:

```
<pu>
<wu wi=1 gv=0 sm="can">"neng2"
<wu wi=2 gv=1 sm="carry out">
  "li3" "xing2"
<wu wi3 gv=1 "contract">
  "he2" "yue4"
</pu>
```

The two structures are different. Fortunately, as no direct reversion of dominance is involved, conversion of parse structures in either direction is possible if the nature of the relevant dependency links (in terms of the grammar formalism used) is available in the annotation.

## 6 Summing up

In our annotation scheme, the basic unit for building syntactic dependency is the word. Below the word, we have morphological units and dictionary entries, and the characters form the terminal text elements.

The adequacy of our annotation scheme can be gauged by comparing it to the “layers” of annotation in the preliminary recommendations on syntactic annotation of corpora of EAGLES (1996): (a) bracketing of segments, (b) labelling the category of segments, (c) showing dependency relations, (d) indicating syntactic function labels, (e) marking subclassification of syntactic segments, (f) logical relations of various kinds, (g) information about the rank of a syntactic unit and (h) information about spoken language non-fluency phenomena. Our annotation scheme includes layers (c) and (d) directly. Layers (a) and (b) are covered by treating dependents (and recursively dependents thereof) of a word as its phrasal projection. Layer (e) is taken care of by a larger-than-minimal tagset and grammatical information stored in the lexical entries, which also help to cover layer (f). As we do not have intermediate phrasal categories in our dependency formalism, layer (g) is not relevant. Layer (h) is not relevant for text corpora.

In our annotation scheme, the sub-word units of morphological units, dictionary units and the terminal text elements (characters) collaborate to take care of morphological complications. We take care to preserve the terminal text elements to enable re-analysis and recovery from errors. Grammatical functions are marked to facilitate conversion for use with other grammatical formulations.

## 7 Conclusion

We have described how we perform syntactic and morphological annotation of a small corpus of Chinese text. The SGML-based annotation scheme has been designed to ensure good tolerance of errors of analysis at various levels and portability to and from other grammatical formalisms and syntactic theories. Future efforts will be in two directions — scaling up with a stylistically more balanced corpus, and importing and adapting annotations performed according to other approaches.

## Acknowledgements

The research reported in this paper is supported by Grant #7000888 and Grant #7001002 of the City University of Hong Kong.

## References

- Sergio Balari and Luca Dini, editors. 1998. *Romance in Head-driven Phrase Structure Grammar*, volume 75 of *CSLI Lecture Notes*. CSLI Publications, Stanford.
- Marie Bourdon, Lyne Da Sylva, Michel Gagnon, Alma Kharrat, Sonja Knoll, and Anna Maclachlan. 1998. A Case Study in Implementing Dependency-Based Grammars. In Alan Polguère and Sylvain Kahane, editors, *Proceedings of COLING-ACL'98 Workshop on Processing of Dependency-Based Grammars*, pages 88–94, Université de Montréal, August.
- Jacques Courtin and Damien Genthial. 1998. Parsing with Dependency Relations and Robust Parsing. In Alan Polguère and Sylvain Kahane, editors, *Proceedings of COLING-ACL'98 Workshop on Processing of Dependency-Based Grammars*, pages 95–101, Université de Montréal, August.
- Michael A. Covington. 1990. Parsing Discontinuous Constituents in Dependency Grammar. *Computational Linguistics*, 16(4):234–236.
- EAGLES. 1996. Recommendations for the Syntactic Annotation of Corpora. EAGLES document EAG-TCWG-SASG/1.8. PISA: Consiglio Nazionale delle Ricerche. Istituto di Linguistica Computazionale. (Authors: G. Leech, R. Barnet, P. Kahrel).
- Haim Gaifman. 1965. Dependency Systems and Phrase-Structure Systems. *Information and Control*, 8:304–337.
- Eva Hajičová. 1991. Free Word Order Described Without Unnecessary Complexity. *Theoretical Linguistics*, 17:99–106.
- David G. Hays. 1964. Dependency Theory: A Formalism and Some Observations. *Language*, 40:511–525.
- Peter Hellwig. 1986. Dependency Unification Grammar. In *Proceedings of 11th International Conference on Computational Linguistics (COLING'86)*, pages 195–199.
- Changning Huang, Chunfa Yuan, and Shimei Pan. 1992. Yuliaoku, Zhishi Huoqu He Jufa Fenxi (Corpora, Knowledge Acquisition and Syntactic Parsing). *Journal of Chinese Information Processing*, 6(3):1–6.
- Richard Hudson. 1984. *Word Grammar*. Blackwell, Oxford.
- Nancy Ide, Greg Priest-Dorman, and Jean Véronis. 1996. Corpus Encoding Standard. Expert Advisory Group on Language Engineering Standards. Last modified 20 March 2000.
- Tom Bong-Yeung Lai and Changning Huang. 1998a. An Approach to Dependency Grammar for Chinese. In Yang Gu, editor, *Studies in Chinese Linguistics*, pages 143–163. Linguistic Society of Hong Kong, Hong Kong.
- Tom Bong Yeung Lai and Changning Huang. 1998b. Complements and Adjuncts in Dependency Grammar Parsing Emulated by a Constrained Context-Free Grammar. In Alan Polguère and Sylvain Kahane, editors, *Proceedings of COLING-ACL'98 Workshop on Processing of Dependency-Based Grammars*, pages 102–108, Université de Montréal, August.
- Tom Bong Yeung Lai and Changning Huang. 1999a. Unification-based Parsing Using Annotated Dependency Rules. In Jost Gippert and Peter Olivier, editors, *Multilinguale Corpora: Codierung, Strukturierung, Analyse -*

11. Jahrestagung der Gesellschaft für Linguistische Daten Verarbeitung (GLDV'99), pages 235–244. Enigma Corporation, Praha, December.
- Tom Bong Yeung Lai and Changning Huang. 1999b. Unification-based Parsing Using Annotated Dependency Rules. In *Proceedings of 5th Natural Language Processing Pacific Rim Symposium 1999 NLPRS'99*, pages 102–107, Beijing, November.
- T.B.Y. Lai, S.C. Lun, C.F. Sun, and M.S. Sun. 1992. A Tagging-Based First-Order Markov Model Approach to Automatic Word Identification for Chinese Sentences. In Julius T. Tou and Joseph J. Liang, editors, *Intelligent Systems for Processing Oriental Languages (Proceedings of the 1992 International Conference on Computer Processing of Chinese and Oriental Languages)*, pages 17–23, Tampa, Florida, December. Chinese Language Computer Society.
- Tom B.Y. Lai, M.S. Sun, S.C. Lun, and B.K. Tsou. 1998. Using Syntactically Motivated Tags in Markov Model Word Segmentation. In *Proceedings of 1998 International Conference on Chinese Information Processing (IC-CIP'98)*, pages 215–222, Beijing.
- Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, and et al. 1994. The Penn. Treebank: annotating predicate argument structure. In *Proceedings of ARPA Speech and Natural Language Workshop*.
- Igor A. Mel'čuk. 1988. *Dependency Syntax: Theory and Practice*. State University of New York Press, New York.
- Makoto Nagao. 1993. Current Status and Further Trends of Natural Language Processing. In *Proceedings of KB&KS'93*, pages 31–34, Tokyo.
- John Nerbonne, Klaus Netter, and Carl J. Pollard, editors. 1994. *German in Head-Driven Phrase Structure Grammar*. Number 46 in CSLI Lecture Notes. CSLI Publications, Stanford University.
- Carl Pollard and Ivan Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago.
- Jane J. Robinson. 1970. Dependency Structures and Transformation Rules. *Language*, 46:259–285.
- SGML. 1986. Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML). Genf: International Organization for Standardization. ISO8897.
- Stanley Starosta. 1988. *The Case for Lexicase*. Pinter Publishers, London.
- Lucien Tesnière. 1959. *Elements de syntaxe structurale*. Klincksieck, Paris.
- Chunfa Yuan and Changning Huang. 1992. Knowledge Acquisition and Chinese Parsing Based on Corpus. In *Proceedings of COLING'92*, pages 13000–13004, Nantes.
- Ming Zhou and Changning Huang. 1994. An Efficient Syntactic Tagging Tool for Corpora. In *Proceedings of COLING'94*, pages 949–955, Kyoto.