A Chinese Dependency Syntax for Treebanking

Haitao Liu, Wei Huang

Department of Applied Linguistics
Communication University of China
CN-100024, Beijing, China
{byliuhaitao, howard}@cuc.edu.cn

Abstract. This paper presents a Chinese dependency syntax for treebanking. The syntax contains 13 word classes and 34 dependency types. A format of treebank based on the syntax is also proposed for the applications of computational and general linguistic research. Some experiments show that the treebank based on the proposed dependency syntax can be used for training and evaluating the dependency parser and for quantitative analysis of Chinese syntax.
Keywords: dependency relation, dependency syntax, treebank, Chinese syntax

1 Introduction

Treebanks are often used as a tool and resource for training and evaluating a syntactic parser in computational linguistics [1]. However, treebanks are not only useful to computational linguists, they are also an important tool for linguists from other branches of linguistics. In this way, if we are planning to build a treebank, perhaps we have to consider its applications in these two fields.

Although the Penn treebank based on phrase structure is still the standard of computational applications of treebank, many new projects, particularly in Europe, like to use dependency structure as the annotation schemes [5].

In this paper, we present a Chinese dependency syntax for treebanking. A format of dependency treebank is proposed for training and evaluate a dependency parser and for quantitative analysis of Chinese.

In section 2, we discuss some kernels of dependency grammar and how to build a dependency grammar for a language. Section 3 presents the proposed Chinese dependency syntax and how to process coordinating structures in this syntax. In section 4, a format of dependency treebank is given and some questions on the treebanking are discussed.

2 Dependency relation and dependency syntax

Sentence analysis based on the dependency relations has a longer history than the method based on phrase structure. The ideas of dependency analysis are found more or less in traditional grammar of many languages. In other words, the school grammars in many countries are similar with the syntactic

principles based on dependency relations. This is an important argument, if we consider that building of treebank is a labor-intensive task.

What's dependency relation, this is not a simple question, because "It is perhaps surprising that one question which, as far as I know, has never been seriously addressed in the Dependency Grammar (DG) literature is what the dependency relation might really amount to" [6]. Considering that Kreps ignores many documents in German and French, which are main publication languages of dependency grammar, we cannot agree completely with him, but he reveals a fact that linguists have still different definitions about this concept.

Although the definition is not unified, but the following properties are generally accepted by linguists to constitute the core properties of a syntactic dependency relation [4][8][10][11][13]:

- It is a binary relation between two linguistic units.
- It is usually asymmetrical, with one of the two units acting as the governor and the other as dependent.
- It is labeled, so the dependency relations should be distinguished and explicitly labeled in the arc linking the two units.

Today Penn treebank is already seen as a gold standard of training and evaluating parser. However, if we want NLP to solve many more language problems for our world, only the Penn treebank is not enough, we have to build more treebanks including more languages and genres, but it is a very difficult task, because "it is unrealistic and almost impossible to train experts in a specialized scientific domain so as to do Penn-style phrase structure annotation to the texts of their expertise." [15] If this is true, we have to search a more easily used scheme to annotate the treebank. Perhaps this also explains why there are more and more treebanks based on dependency relations.

It is worthwhile to notice that, although we have some native dependency treebanks, such as PDT (Prague Dependency Treebank, [3]), many studies about dependency parsing are using a treebank converted from the Penn treebank. It is understandable to use such non-native dependency for easily comparing the results with other methods. Nevertheless, this approach is problematic, because the fundamental principles of sentence analysis in these two methods are not completely the same. For example, there are too few dependency types in the non-native dependency treebank, which is not acceptable for dependency syntactician.

How to build a dependency syntax based on the concepts of dependency grammar tradition? Firstly, the linguist has to determine the word classes of the defined language. Secondly, he should define the dependency types (relations) of a language.

After having a set of word class and dependency types, we can construct dependency syntax of a language, which diagrammatically is presented as in figure 1. where wc_1 wc_2...wc_m are word classes of the language and dep_1 dep_2...dep_n are the dependency types of the language. Figure 1 gives not only the capacity of a word class governing other word classes through some dependency types, it also provides a capacity of a word class governed by other word classes. These two capacities consist of the general valency pattern of a word class and the pattern is the driven force of generating the dependency relations. Based on the valency pattern, for improving the expressing and explaining capacity of the valency patterns, a new kind of pattern with probabilistic elements is developed. Under the new name "Weighted

Valency Pattern", the relation between two word classes will not only be described qualitatively, but also be defined quantitatively [8].

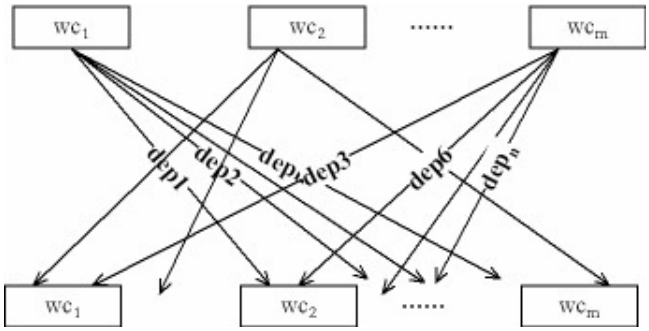

Fig. 1. Diagrammatic dependency syntax

If word class and dependency type are the core elements of dependency syntax, what is the number of the word classes and types in a language? If it is too few, the analysis and learning based on the syntax will be easier, but it will degrade the describing precision of the model. If it were too numerous, the recognizing of the type will be more difficult for human and machine, and turns down the precision and efficiency of annotating and parsing. For finding an appropriate value, we investigate dependency grammars of more than 10 languages. The table 1 shows the result[1].

Table 1. The amount of word class and dependency type of some languages

Language	Word class	Complements	Adjuncts
English	11	14	11
German	10	17	9
Danish	11	9	6
Polish	10	10	8
Bangla	14	10	10
Finnish	14	12	9
Hungarian	10	11	10
Japanese	19	13	7
Esperanto	12	8	10
French	10	11	10
Italian	9	10	7
Chinese	11	19	16

The table shows that there are often about 10-20 word classes and 15-35 dependency types in the dependency syntax of a language.

[1] The data are extracted from [9] and other technical report published by BSO/DLT project. The amount of dependency type is the sum of complements and adjuncts.

3 Chinese dependency syntax

Dependency syntax of a language contains two parts: the tagset of word classes, the tagset of dependency types. Our task is for building a dependency syntax to annotate a treebank, which is a time-consuming and labor-intensive task, we have to think how to automate the process and find more competent workers from early beginning. In other words, we should use the available resource as far as possible.

Based on the national standard “POS tagset for Chinese information processing" (2003) and popularly used “Grammar system for middle-school teaching”, we propose a set of word class with 13 types: noun (n), verb (v), adjective (a), adverb (d), pronoun (r), preposition (p), numeral (m), classifier (q), conjunction(c), interjection (e), particle (u), onomatopoeia (o) and punctuation (bd). Compared with the national standard of POS tags, we remove some tags, which do not work on the level of syntax. Comparatively with traditional school grammar, we gives some functional (particle) words an important position in the syntax, because they often play a crucial role during determining the dependency relation between two words. Figure 2 shows a hierarchy of POS tagset in Chinese[2].

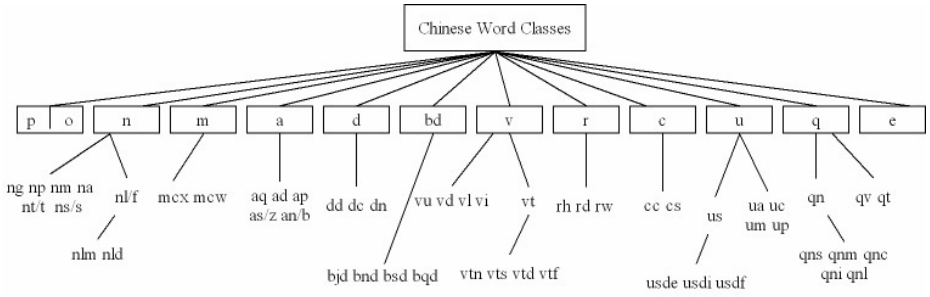

Fig. 2. A hierarchy of word classes in Chinese

According to the three basic elements consisting of dependency relation: governor, dependent and dependency type, the tagset of Chinese dependency type is built.

Table 2. Chinese dependency types

Type	Label	Type	Label
Main governor	S	Sentential object	SentObj
Subject	SUBJ	Auxiliary verb	ObjA
Object	OBJ	Coordinating mark	C-
Indirect Object	OBJ2	Adverbial	AVDA
Subobject	SUBOBJ	Verb adjunct	VA
Subject Complement	SOC	Attributer	ATR
Prepositional Object	POBJ	Topic	TOP
Postpositional Complement	FC	Coordinating adjunct	COOR

[2] [7] includes a detail explanation about subclass’s tags.

Complement	COMP	Epithet	EPA
Complement of usde ‘的’	DEC	Numeral adjunct	MA
Complement of usdi ‘地’	DIC	Aspect adjunct	TA
Complement of usdf ‘得’	DFC	Adjunct of sentence end	ESA
Object of Pba ‘把’	BaOBJ	Parenthesis	InA
Plural complement	PLC	Clause adjunct	CR
Ordinal complement	OC	Correlative adjunct	CsR
Complement of classifier	QC	Particle adjunct	AuxR
Construction of Pbei ‘被’	BeiS	Punctuation	Punct

The dependency tagset contains 20 complements and 14 adjuncts. The amount of Chinese complements is a little more than other languages, because Chinese has to use the functional words for completing the grammatical functions, which often are morphologically realized in other languages.

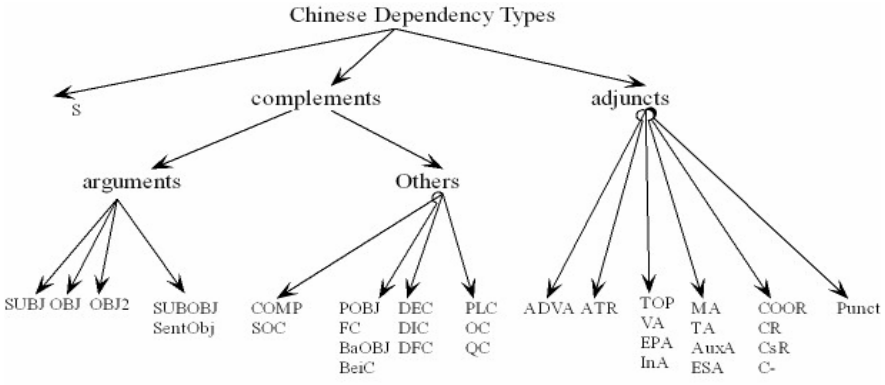

Fig. 3. A hierarchy of dependency types in Chinese

We use a method similar to PDT [16] and DLT [9] to process the coordinating structure. In Chinese the often used conjunctions are *he*(和), *yu*(与), *tong*(同) and punctuation(、). Figure 3 shows a structure with 2 conjunctions and three conjuncts.

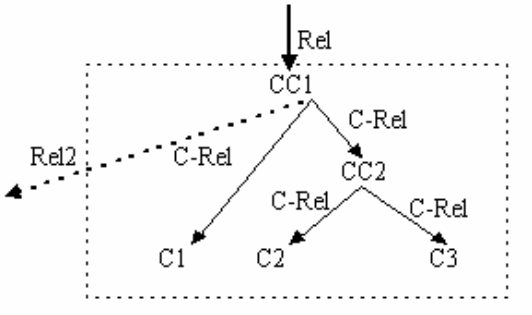

Fig. 4. Coordinating structure

In a coordinating structure, the first conjunction works as the head of the whole structure to connect with the head and dependents of the other structures. The prefix C- is introduced for replicating the de-

pendency type which governs the whole structure. For example, in the sentence “美丽的长江、黄河和黑龙江都是中国的河流”(The beautiful Yangzi, Huanghe and Heilongjiang are the rivers of China), the Chinese character string “长江 、 黄河 和 黑龙江” forms a coordinating structure. Where Rel-SUBJ, Rel2-ATR, CC1-‘、’, CC2-和, C-Rel-C-SUBJ, C1-长江, C2-黄河, C3-黑龙江.

4 Chinese dependency treebank

After having the dependency syntax, we have adopted the format in table 2 for a Chinese dependency treebank.

Table 3. Annotation of a sample sentence in treebank

Order number of Sentence	Word			Governor			Dependency type
	Order number	Character	POS	Order number	Character	POS	
S1	1	这	r	2	是	v	subj
S1	2	是	v	6	。	bjd	s
S1	3	三	m	4	个	q	qc
S1	4	个	q	5	例子	n	atr
S1	5	例子	n	2	是	v	obj
S1	6	。	bjd				

This format includes all three mentioned properties of dependency relation, but sometimes it is also helpful to construct a connected directed labeled graph [10] as in Figure 5.

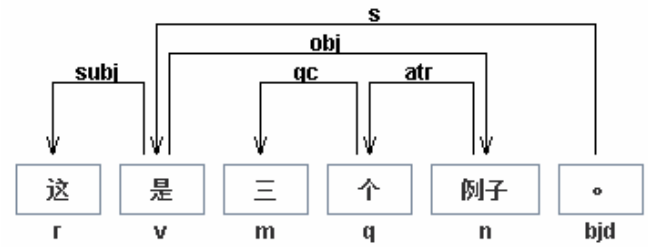

Fig. 4. The dependency analysis as a graph

It seems that the proposed format includes some redundant information only from the viewpoint of computational linguistics, but because we also hope to use the treebank for quantitative analysis of Chinese, current format is helpful to satisfy the mentioned two tasks. It is easy to convert the format into a more exchangeable XML format[3].

- <sentence id="**1**" user=“**Liu Haitao**" date="**2006-06-22**">
  <word id="**1**" form="**这**" postag="**r**" head="**2**" deprel="**subj**" />
  <word id="**2**" form="**是**" postag="**v**" head="**6**" deprel="**s**" />

[3] This is a Malt-XML format. http://w3.msi.vxu.se/~nivre/research/MaltXML.html

<word id="3" form="三" postag="m" head="4" deprel="qc" />
<word id="4" form="个" postag="q" head="5" deprel="atr" />
<word id="5" form="例子" postag="n" head="2" deprel="obj" />
<word id="6" form="。" postag="bjd" head="0" deprel="ROOT" />
</sentence>

For verifying the dependency syntax and the format, we have built a tentative dependency treebank[4]. The treebank is built on the news (xinwen lianbo) of China Central Television, a genre which is intended to be spoken but whose style is similar to the written language. We select four complete broadcasting news of this program as the annotated material. For finding the comfortable annotation means, we make two groups of annotator to annotate the texts. Before beginning the annotation, training about dependency grammar, how to recognize a dependency relation in particular, was given to all participators of this project. The first group consists of 20 undergraduates of Chinese linguistics, and the other group is only a graduate of computational linguistics with a background of Chinese linguistics. While everyone of the first group annotates the text with 500 tokens, the graduate has to finish the text with 10000 tokens. Then, two teachers of Chinese linguistics review all annotated texts. The final treebank includes 711 sentences and 20034 word tokens, so the mean sentence length is 28 words.

The annotators use MS-Excel or Access as the tool. It is feasible, because our treebank is very small, but these tools are not suitable for annotating long sentence. For making the annotation more easily, a system, with the name Depenendecy Grammar Annotator (DGA), has been developed [17].

Compared with the Penn Chinese Treebank [14] and the Sinica Treebank [2], our treebank is a native dependency treebank, whose annotate scheme is mainly based on the tradition of dependency grammar without phrase structure. We are using a pure syntactic annotation scheme, which make a distinction with the Prague Dependency Treebank [3] and the PropBank [12].

We have also successfully made some experiments based on the treebank, for instance, the dependency parsing and quantitative analysis of Chinese A statistical Chinese dependency parser, using a general parser Maltparser released by Nivre [11] and the treebank built in this paper, has got the score 0.759 (UAS, unlabeled attachment score) and 0.712 (LAS, labeled attachment score). The analysis based on the treebank shows that Chinese has much in common with other languages, for example, the distribution of noun in a text; but it also shows some particularities, for example that dependency distance is much greater in Chinese than in English, German and Japanese [8].

We are annotating a spoken language corpus using the same syntax and scheme, that is a necessary complementation to the current treebank.

5 Conclusions

In the paper, we propose a Chinese dependency syntax and a format of treebank based on the syntax. A tentative dependency treebank is built for verifying that the proposed syntax is suitable to annotate a Chinese text. The built treebank is not only used as the resource to train and evaluate the parser, also

[4] Annotation guidelines in [7].

works as a database for quantitative analysis of Chinese. The experiments show that the proposed dependency syntax and treebank are useful not only for computational linguistics, but it is also helpful for general syntactic studies.

Acknowledgments. We thank Richard Hudson and Joakim Nivre for detailed comments on an earlier stage of this paper. Work described in this paper was partly supported by The State Administration of Radio, Film & TV of China (the research project BW0357) and Communication University of China (the research project BBU211-15.4).

References

1. Abeillé, Anne (ed. 2003) *Treebank: Building and using Parsed Corpora.* Dordrecht: Kluwer.
2. Chen, Keh-Jiann et al. (2003) Sinica Treebank: Design Criteria, Representational Issues and Implementation. in Anne Abeille (ed. 2003). pp. 231-248.
3. Hajič, Jan, Alena Böhmová, Eva Hajičová, Barbora Vidová Hladká (2003) The Prague Dependency Treebank: A Three-Level Annotation Scenario. in A. Abeillé (ed. 2003), pp. 103-127.
4. Hudson, R. A. (1990) *English Word Grammar*. Oxford: Blackwell.
5. Kakkonen, T. (2005) Dependency Treebanks: Methods, Annotation Schemes and Tools. in *Proceedings of the 15th Nordic Conference of Computational Linguistics* (NODALIDA 2005). Joensuu, Finland.
6. Kreps, Christian (1997) *Extraction, Movement and Dependency Theory*. PhD thesis, University College London.
7. Liu, Haitao (2005) A Chinese syntax based on dependency relations. CUC Tech. Report.
8. Liu, Haitao (2006) Syntactic parsing based on Dependency Relations. in *grkg/Humankybernetik*, 47(3): 124-135.
9. Maxwell, D. and Schubert, K. (1989) *Metataxis in practice: dependency syntax for multilingual machine translation*. Dordrecht: Foris.
10. Mel'cuk, I. A. (1988) *Dependency syntax: theory and practice*. Albany: State University Press of New York.
11. Nivre, J. (2006) *Inductive Dependency Parsing*. Dordrecht: Springer.
12. Palmer, Martha, Dan Gildea, and Paul Kingsbury (2005) The Proposition Bank: A Corpus Annotated with Semantic Roles, *Computational Linguistics*, 31(1): 71-106.
13. Tesnière, L. (1959) *Eléments de la syntaxe structurale.* Paris: Klincksieck.
14. Xue, Nianwen, Fei Xia, Fu-Dong Chiou, and Martha Palmer (2005) The Penn Chinese TreeBank: Phrase Structure Annotation of a Large Corpus. *Natural Language Engineering*, 11(2): 207-238.
15. Yamada, Hiroyasu and Yuji Matsumoto (2003) Statistical dependency analysis with Support Vector Machines. in *Proc. 8th International Workshop on Parsing Technologies (IWPT)*, pp.195-206, April 2003.
16. Žabokrtský, Zdeněk (2005) *Valency Lexicon of Czech Verbs.* Ph.D. Thesis, Faculty of Mathematics and Physics, Charles University in Prague.
17. Zuo, Wei (2005) *Implementation of Dependency Grammar Annotator.* Master thesis, Communication University of China.