Recovering Non-Local Dependencies for Chinese

Yuqing Guo
NCLT, School of Computing
Dublin City University
Dublin 9, Ireland
yguo@computing.dcu.ie

Haifeng Wang
Toshiba (China)
Research and Development Center
Beijing, 100738, China
wanghaifeng@rdc.toshiba.com.cn

Josef van Genabith
NCLT, School of Computing
Dublin City University
IBM CAS, Dublin, Ireland
josef@computing.dcu.ie

Abstract

To date, work on Non-Local Dependencies (NLDs) has focused almost exclusively on English and it is an open research question how well these approaches migrate to other languages. This paper surveys non-local dependency constructions in Chinese as represented in the Penn Chinese Treebank (CTB) and provides an approach for generating proper predicate-argument-modifier structures including NLDs from surface context-free phrase structure trees. Our approach recovers non-local dependencies at the level of Lexical-Functional Grammar f-structures, using automatically acquired subcategorisation frames and f-structure paths linking antecedents and traces in NLDs. Currently our algorithm achieves 92.2% f-score for trace insertion and 84.3% for antecedent recovery evaluating on gold-standard CTB trees, and 64.7% and 54.7%, respectively, on CTB-trained state-of-the-art parser output trees.

1 Introduction

A substantial number of linguistic phenomena such as topicalisation, relativisation, coordination and raising & control constructions, permit a constituent in one position to bear the grammatical role associated with another position. These relationships are referred to Non-Local Dependencies (NLDs), where the surface location of the constituent is called "antecedent", and the site where the antecedent should be interpreted semantically is called "trace". Capturing non-local dependencies is crucial to the accurate and complete determination of semantic interpretation in the form of predicate-argument-modifier structures or deep dependencies.

However, with few exceptions (Model 3 of Collins, 1999; Schmid, 2006), output trees produced by state-of-the-art broad coverage statistical parsers (Charniak, 2000; Bikel, 2004) are only surface context-free phrase structure trees (CFG-trees) without empty categories and coindexation to represent displaced constituents. Because of the importance of non-local dependencies in the proper determination of predicate-argument structures, recent years have witnessed a considerable amount of research on reconstructing such hidden relationships in CFG-trees. Three strategies have been proposed: (i) post-processing parser output with pattern matchers (Johnson, 2002), linguistic principles (Campbell, 2004) or machine learning methods (Higgins, 2003; Levy and Manning, 2004; Gabbard et al., 2006) to recover empty nodes and identify their antecedents;[1] (ii) integrating non-local dependency recovery into the parser by enriching a simple PCFG model with GPSG-style gap features (Collins, 1999; Schmid, 2006); (iii) pre-processing the input sentence with a finite-state trace tagger which detects empty nodes before parsing, and identify the antecedents on the parser output with the gap information (Dienes and Dubey, 2003a; Dienes and Dubey, 2003b).

In addition to CFG-oriented approaches, a number of richer treebank-based grammar acquisition and parsing methods based on HPSG (Miyao et al., 2003), CCG (Clark and Hockenmaier, 2002), LFG (Riezler et al., 2002; Cahill et al., 2004) and Dependency Grammar (Nivre and Nilsson, 2005) incorporate non-local dependencies into their deep syntactic or semantic representations.

A common characteristic of all these approaches

[1] (Jijkoun, 2003; Jijkoun and Rijke, 2004) also describe post-processing methods to recover NLDs, which are applied to syntactic dependency structures converted from CFG-trees.

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 257–266, Prague, June 2007.

is that, to date, the research has focused almost entirely on English,[2] despite the disparity in type and frequency of non-local dependencies for various languages. In this paper, we address recovering non-local dependencies for Chinese, a language drastically different from English and whose special features such as lack of morphological inflection make NLD recovery more challenging. Inspired by (Cahill et al., 2004)'s methodology which was originally designed for English and Penn-II treebank, our approach to Chinese non-local dependency recovery is based on Lexical-Functional Grammar (LFG), a formalism that involves both phrase structure trees and predicate-argument structures. NLDs are recovered in LFG f-structures using automatically acquired subcategorisation frames and finite approximations of functional uncertainty equations describing NLD paths at the level of f-structures.

The paper is structured as follows: in Section 2 we outline the distinguishing features of Chinese non-local dependencies compared to English. In Section 3 we review (Cahill et al., 2004)'s method for recovering English NLDs in treebank-based LFG approximations. In Section 4, we describe how we modify and substantially extend the previous method to recover all types of NLDs for Chinese data. We present experiments and provide a dependency-based evaluation in Section 5. Finally we conclude and summarise future work.

2 Non-Local Dependencies in Chinese

In the Penn Chinese Treebank (CTB) (Xue et al., 2002) non-local dependencies are represented in terms of empty categories (ECs) and (for some of them) coindexation with antecedents, as exemplified in Figure 1. Following previous work for English and the CTB annotation scheme, we use "non-local dependencies" as a cover term for all missing or dislocated elements represented in the CTB as an empty category (with or without coindexation/antecedent), and our use of the term remains agnostic about fine-grained distinctions between non-local dependencies drawn in the theoretical linguistics literature.

In order to give an overview on the characteristics of Chinese non-local dependencies, we extracted all empty categories together with coindexed antecedents from the Penn Chinese Treebank version 5.1 (CTB5.1). Table 1 gives a breakdown of the most frequent types of empty categories and their antecedents, which account for 43,791 of the total 43,954 (99.6%) ECs in CTB5.1.[3]

According to their different linguistics properties, we divide the empty nodes listed in Table 1 into three major types: null relative pronouns, locally mediated dependencies, and long-distance dependencies.

Null Relative Pronouns (lines 2, 7) themselves are local dependencies, and thus are not coindexed with an antecedent. But they mediate non-local dependencies by functioning as antecedents for the dis-

(1) 不 愿意 发掘 培植 有 潜力 的 新 作家
not want look-for train have potential DE new writer
'(People) don't want to look for and train new writers who have potential.'

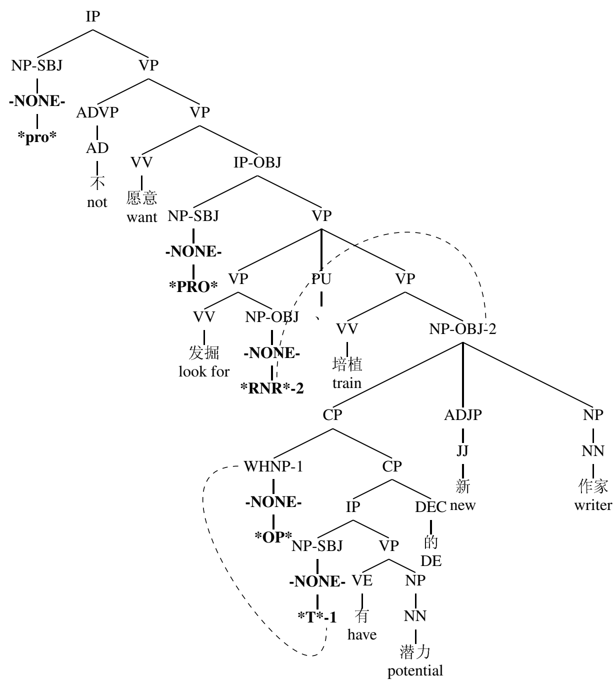

Figure 1: Example of non-local annotations in CTB, including dropped subject (*pro*), control subject (*PRO*), relative clause (*T*), and coordination (*RNR*).

[2] (Levy and Manning, 2004) is the only approach we are aware of that has been applied to both English and German.

[3] An extensive description of the types of empty categories and the use of coindexation in CTB can be found in Section VI of the bracketing guidelines.

	Antecedent	POS	Label	Count	Description
1	WHNP	NP	*T*	11670	WH trace (e.g. *OP*中国/China发射/launch***T***的/DE卫星/satellite)
2		WHNP	*OP*	11621	Empty relative pronouns (e.g. ***OP***中国/China发射/launch的/DE卫星/satellite)
3		NP	*PRO*	10946	Control constructions (e.g. 这里/here不/not许/allow***PRO***抽烟/smoke)
4		NP	*pro*	7481	Pro-drop situations (e.g. ***pro***不/not曾/ever遇到/encounter的/DE问题/problem)
5	IP	IP	*T*	575	Topicalisation (e.g. 我们/we能/can赢/win，他/he说/say***T***)
6	WHPP	PP	*T*	337	WH trace (e.g. *OP*人口/population***T***密集/dense地区/area)
7		WHPP	*OP*	337	Empty relative pronouns (e.g. ***OP***人口/population密集/dense地区/area)
8	NP	NP	*	291	Raising & passive constructions (e.g. 我们/we被/BEI排除/exclude*****在外/outside)
9	NP	NP	*RNR*	258	Coordinations (e.g. 鼓励/encourage***RNR***和/and支持/support投资/investment)
10	CLP	CLP	*RNR*	182	Coordinations (e.g. 五/five***RNR***至/to十/ten亿/hundred million元/Yuan)
11	NP	NP	*T*	93	Topicalisation (e.g. 薪水/salary都/all用/use***T***来/for享乐/pleasure)

Table 1: The distribution of the most frequent types of empty categories and their antecedents in CTB5.1. The types with frequency less than 30 are ignored.

located constituent inside a relative clause.[4]

Locally Mediated Dependencies are non-local as they are projected through a third lexical item (such as a control or raising verb) which involves a dependency between two adjacent levels and they are therefore bounded. This type encompasses: (line 8) raising constructions, and short-bei constructions (passivisation); (line 3) control constructions, which includes two different types: a generic *PRO* with an arbitrary reading (approximately equals to unexpressed subjects of *to*-infinitive and gerund verbs in English); and a *PRO* with definite reference (subject or object control).[5]

Long-Distance Dependencies (LDDs) differ from locally mediated dependencies, in that the path linking the antecedent and trace might be unbounded (also called unbounded, long-range dependencies). LDDs include the following phenomena:

Wh-traces in relative clauses, where an argument (line 1) or adjunct (line 6) "moves" and is coindexed with the "extraction" site.

Topicalisation (lines 5, 11) is one of the typical LDDs in English, whereas in Chinese not all topics involve displacement, for instance (2).

(2) 北京 秋天 最 美
Beijing autumn most beautiful
'Autumn is the most beautiful in Beijing.'

Coordination is divided into two groups: right node raising of an NP phrase which is an argument shared by the coordinate predicates (line 9); and the coordination of quantifier phrases (line 10) and verbal phrases (3), in which the antecedent and trace are both predicates and possibly take their own arguments or adjuncts.

(3) 我 和 他 分别 去 公司 和 *RNR* 医院
I and he respectively go to company and *RNR* hospital
'I went to the company and he went to the hospital respectively.'

Pro-drop situations (line 4) are prominent in Chinese because subject and object are only semantically but not syntactically required. Nevertheless we also treat pro-drop as a long-distance dependency as in principle the dropped subjects can be determined from the general (often inter-sentential) context.

Table 2 gives a quantitative comparison of NLDs between Chinese data in CTB5.1 and English in Penn-II. The data reveals that: first, NLDs in Chinese are much more frequent than in English (by nearly 1.5 times); and moreover 69% are not explicitly linked to an antecedent, compared to 43% for English, due to the high prevalence of pro-drop in Chinese.

	# of sent	# of EC	# of EC/sent	# non-coindex	% non-coindex
Chinese	18,804	43,954	2.34	30,429	69.23
English	49,207	79,245	1.61	34,455	43.48

Table 2: Comparison of NLDs between Chinese data in CTB5.1 and English in Penn-II .

[4]Null relative pronouns used in the CTB annotation are to distinguish relative clauses in which an argument or adjunct of the embedded verb is "missing" from complement (appositive) clauses which do not involve non-local dependencies.

[5]However in this case the CTB annotation doesn't coindex the locus (trace) with its controller (antecedent).

(4) 钱 我们 用 来 享乐
money we use to please
'Money, we use for pleasure.'

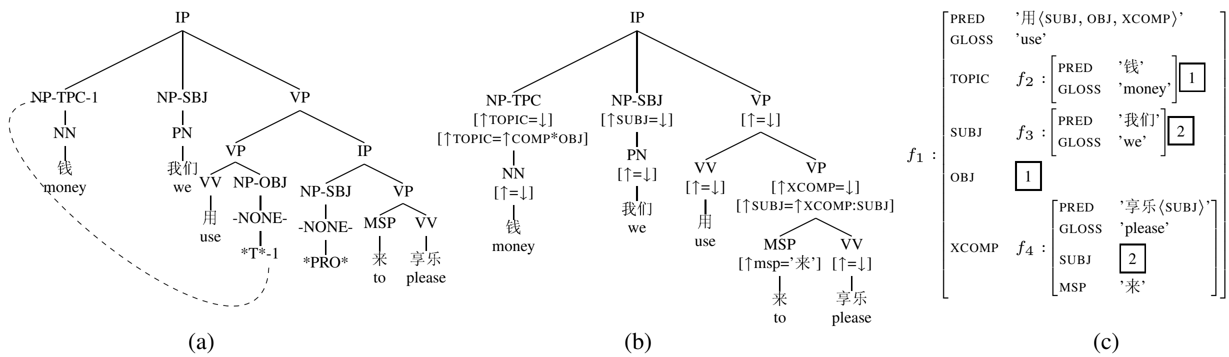

Figure 2: (a) the CTB tree; (b) LFG c-structure with functional equations; (c) corresponding f-structure. (↑) in the functional annotation refers to the f-structure associated with the mother node and (↓) to that of the local node.

3 NLD Recovery in LFG Approximations

3.1 Lexical Functional Grammar

Lexical Functional Grammar (Kaplan and Bresnan, 1982) is a constraint-based grammar formalism which minimally involves two levels of syntactic representation: c(onstituent)-structure and f(unctional)-structure. C-structure takes the form of CFG-trees and captures surface grammatical configurations. F-structure encodes more abstract grammatical functions (GFs) such as SUBJ(ect), OBJ(ect), COMP(lement), ADJ(unct) and TOPIC etc., in the form of Attribute Value Matrices which approximate to basic predicate-argument-adjunct structures or dependency relations. C-structures are related to f-structures by functional annotations (cf. Figure 2 (b) & (c)).

In LFG, non-local dependencies are captured at f-structure level in terms of reentrancies, indicated $\boxed{1}$ for the topicalisation and $\boxed{2}$ for the control construction in Figure 2(c) obviating the need for traces and coindexation in the c-structure (Figure 2(b)), unlike in CTB trees (Figure 2(a)). LFG uses functional uncertainty (FU) equations (regular expressions) to specify paths in f-structures between the trace and its antecedent. To account for the reentrancy $\boxed{1}$ in the f-structure, a FU equation of the form ↑TOPIC=↑COMP*OBJ is required (as the length of the dependency might be unbounded). The equation states that the value of the TOPIC attribute is token identical with the value of the final OBJ argument along a path through the immediately enclosing f-structure along zero or more COMP attributes.

In addition to FU equations, subcategorisation information is also a significant ingredient in LFG's account of non-local dependencies. Subcategorisation frames (subcat frames) specify the governable grammatical functions (i.e. arguments) required by a particular predicate. In Figure 2(c) each predicate in the f-structure is followed by its subcat frame.

3.2 F-Structure Based NLD Recovery

(Cahill et al., 2004) presented a NLD recovery algorithm operating at LFG f-structure for treebank-based LFG approximations. The method automatically converts Penn-II treebank trees with traces and coindexation into proper f-structures where traces and coindexation in treebank trees (Figure 2(a)) are represented as corresponding reentrances in f-structures (Figure 2(c)), and from the f-structures automatically extracts subcat frames by collecting all arguments of the local predicate at each level of the f-structures, and further acquires finite approximations of FU equations by extracting paths linking the reentracies occurring in the f-structures.

(Cahill et al., 2004)'s approach for English resolves three LDD types in parser output trees without traces and coindexation (Figure 2(b)), i.e. topicalisation (TOPIC), wh-movement in relative clauses (TOPIC_REL) and interrogatives (FOCUS). Given a set of subcat frames s for lemma w with probabilities $P(s|w)$, a set of paths p linking reentrancies conditioned on the triggering antecedent a (TOPIC, TOPIC_REL or FOCUS) with probabilities $P(p|a)$, the core algorithm recursively traverses an f-structure f to:

- find a TOPIC|TOPIC_REL|FOCUS:g pair;
- traverse f along path p to the sub-f-structure h;
- retrieve the local PRED:w at h, and insert g to h iff
  * all GFs specified in the subcat frame s except g are present at h (completeness condition)
  * no other governable GFs present at h are specified in s (coherence condition)
- rank resolution candidates according to the product of subcat frame and NLD path probabilities (Eq. 1).

$$P(s|w) \times P(p|a) \tag{1}$$

4 NLD Recovery Algorithm for Chinese

4.1 Automatic F-Structure Generation

Our NLD recovery is done at the level of LFG f-structures. Inspired by (Cahill et al., 2004; Burke et al., 2004), we have implemented an f-structure annotation algorithm to automatically obtain f-structures from CFG-trees in the CTB5.1. The f-structure annotation algorithm, described below, is applied both to the original CTB trees providing functional tags, traces and coindexation to generate the training corpus, and to the parser output trees without traces and coindexation to provide the f-structure input for NLD recovery.

1. The CFG-trees are head-lexicalised by head-finding rules similar to (Collins, 1999), adapted to CTB.
2. Each local subtree of depth one is partitioned by the head into left and right context. Left-right context rules exploiting configurational, categorial and CTB functional tag information are used to assign each left and right constituent with appropriate functional equations.
3. Empty nodes and coindexation in the CTB trees are automatically captured into corresponding reentrances at f-structure via functional equations.
4. All the functional equations are collected and then passed to a constraint solver to generate f-structures.

4.2 Adaptation to Chinese

(Cahill et al., 2004)'s algorithm (Section 3.2) only resolves certain NLDs with known types of antecedents (TOPIC, TOPIC_REL and FOCUS) at f-structures. However, as illustrated in Section 2, except for relative clauses, the antecedents in Chinese NLDs do not systematically correspond to types of grammatical function. Furthermore nearly 70% of all empty categories are not coindexed with an antecedent. In order to resolve all Chinese NLDs represented in the CTB, we modify and substantially extend the (Cahill et al., 2004) (henceforth C04 for short) algorithm as follows:

Given the set of subcat frames s for the word w, and a set of paths p for the trace t, the algorithm traverses the f-structure f to:

- predict a dislocated argument t at a sub-f-structure h by comparing the local PRED:w to w's subcat frames s
- t can be inserted at h if h together with t is complete and coherent relative to subcat frame s
- traverse f starting from t along the path p
- link t to it's antecedent a if p's ending GF a exists in a sub-f-structure within f; or leave t without an antecedent if an empty path for t exists

In the modified algorithm, we condition the probability of NLD path p (including the empty path without an antecedent) on the GF associated of the trace t rather than the antecedent a as in C04. The path probability $P(p|t)$ is estimated as:

$$P(p|t) = \frac{count(p,t)}{\sum_{i=1}^{n} count(p_i,t)} \tag{2}$$

In contrast even to English, Chinese has very little morphological information. As a result, every word in Chinese has a unique form regardless of its syntactic distribution. For this reason we use more syntactic features w_feats in addition to word form to discriminate between appropriate subcat frames s. For a given word w, w_feats include:

- w_pos: the part-of-speech of w
- w_gf: the grammatical function of w

$P(s|w, w\_feats)$ replaces C04's $P(s|w)$ as lexical subcat frame probability and is estimated as:

$$P(s|w, w\_feats) = \frac{count(s, w, w\_feats)}{\sum_{i=1}^{n} count(s_i, w, w\_feats)} \quad (3)$$

As more conditioning features may cause sever sparse-data problems, in order to increase the coverage of the automatically acquired subcat frames, the subcat frame frequencies $count(s, w, w\_feats)$ are smoothed by backing off to w's part-of-speech w_pos according to Eq. (4). $P(s|w\_pos)$ is estimated according to Eq. (5) and weighted by a parameter Θ. The lexical subcat frame probabilities are estimated from the smoothed frequencies as shown in Eq. (6).

$$count_{bk}(s, w, w\_feats) = count(s, w, w\_feats) + \Theta P(s|w\_pos) \quad (4)$$

$$P(s|w\_pos) = \frac{count(s, w\_pos, w\_gf)}{\sum_{i=1}^{n} count(s_i, w\_pos, w\_gf)} \quad (5)$$

$$P_{bk}(s|w, w\_feats) = \frac{count_{bk}(s, w, w\_feats)}{\sum_{i=1}^{n} count_{bk}(s_i, w, w\_feats)} \quad (6)$$

Finally, NLD resolutions are ranked according to:

$$P_{bk}(s|w, w\_feats) \times \prod_{j=1}^{m} P(p|t_j) \quad (7)$$

As, apart from the maximum number of arguments in a subcat frame, there is no a priori limit on the number of dislocated arguments in a local f-structure, we rank resolutions with the product of the path probabilities of each (of m) missing argument(s).

4.3 A Hybrid Fine-Grained Strategy

As described in Section 2, there are three types of NLDs in the CTB, and their different linguistic properties may require fine-grained recovery strategies. Furthermore, as the NLD recovery method described in Section 4.2 is triggered by "missing" subcategorisable grammatical functions, a few cases of NLDs in which the trace is not an argument in the f-structure, e.g. an ADJUNCT or TOPIC in relative clauses or an null PRED in verbal coordination, can not be recovered by the algorithm. Table 3 shows the types of NLD that can be recovered by C04 and by the algorithm presented in Section 4.2. Table 3 shows that a hybrid methodology is required to resolve all types of NLDs in the CTB. The hybrid method involves four strategies:

- Applying a few simple heuristic rules to insert the empty PRED for coordinations and null relative pronouns for relative constructions. The former is done by comparing the part-of-speech of the local predicates and their arguments in each coordinate; and the latter is triggered by GF ADJUNCT_REL in our system.
- Inserting an empty node with GF SUBJ for short-bei construction, control and raising constructions, and relate it to the upper-level SUBJ or OBJ accordingly.
- Exploiting the C04 algorithm to resolve the wh-trace in relativisation, including ungovernable GFs TOPIC and ADJUNCT.
- Using our modified algorithm (Section 4.2) to resolve the remaining types, viz. long-distance dependencies in Chinese.

	Antecedent			Trace	
	Topic_Rel	Other	Null	Argument	Adjunct
C04	√			√	√
Ours	√	√	√	√	

Table 3: Comparison of the ability of NLD recovery for Chinese between C04 and our algorithm

5 Experiments and Evaluation

For all our experiments, we used the first 760 articles (chtb_001.fid to chtb_931.fid, 10,384 sentences) of CTB5.1, from which 75 double-annotated files (chtb_001.fid to chtb_043.fid and chtb_900.fid to chtb_931.fid, 1,046 sentences) were used as test data,[6] 75 files (chtb_306.fid to chtb_325.fid and chtb_400.fid to chtb_454.fid, 1,082 sentences) were held out as development data, while the other 610 files (8,256 sentences) were used as training data. Experiments were carried out on two different kinds of input: first on CTB gold standard trees stripped of all empty nodes and coindexation information; and

[6]The complete list of double-annotated files can be found in the documentation of CTB5.1.

second, on the output trees of Bikel's parser (Bikel, 2004).

The evaluation metric adopted by most previous work used the label and string position of the trace and its antecedent (Johnson, 2002). As pointed out by (Campbell, 2004), this metric is insensitive to the correct attachment of the EC into the parse tree, and more importantly it is not clear whether it adequately measures performance in predicate-argument structure recovery. Therefore, we use a predicate-argument based evaluation method instead. The NLD recovery is represented as a triple in the form of $\text{REL}(\text{PRED} : loc, \text{GF} : loc)$, where REL is the relation between the dislocated GF and the PRED. In the evaluation for insertion of traces, the GF is represented by the empty category, and in the evaluation for antecedent recovery, the GF is realised by the predicate of the antecedent, e.g. OBJ(用/use:3, 钱/money:1) in Figure 2(c). The antecedent and PRED are both numbered with their string position in the input sentence. Precision, recall and f-score are calculated for the evaluation.

5.1 CTB-Based F-Structure and NLD Resources Acquisition

5.1.1 Automatically Acquired F-Structures

As described in Section 4.1, we automatically generate LFG f-structures from the CTB trees to obtain the training data and generate f-structures from the parser output trees, on which the NLDs will be recovered. To evaluate the performance of the automatic f-structure annotation algorithm, we randomly selected 200 sentences from the test set and manually annotated the f-structures to generate a gold standard. The evaluation metric is the same as for NLD recovery in terms of predicate-argument relations. Table 4 reports the results against the 200-sentence gold standard given the original CTB trees and trees output by Bikel's parser.

Dependencies	Precision	Recall	F-Score
CTB Trees	95.60	95.82	95.71
Parser Output	74.37	73.15	73.75

Table 4: Evaluation of f-structure annotation

5.1.2 Acquiring Subcat Frames and NLD Paths

From the automatically generated f-structure training data, we extract 144,119 different lexical subcat frames and 178 paths linking traces and antecedents for NLD recovery. Tables 5 & 6 show some examples of the automatically extracted subcat frames and NLD paths respectively.

Word:POS-GF(Subcat Frames)	**Prob.**
创立:VV-adj_rel([subj,obj])	0.7655
创立:VV-adj_rel([subj])	0.1537
创立:VV-adj_rel([subj,xcomp])	0.0337
......	...
创立:VV-coord([subj,obj])	0.7915
创立:VV-coord([subj])	0.0975
......	...
创立:VV-top([subj,obj])	0.5247
创立:VV-top([subj,comp])	0.2077
......	...

Table 5: Examples of subcat frames

Trace (Path)	**Prob.**
adjunct(up-adjunct:down-topic_rel)	0.9018
adjunct(up-adjunct:up-coord:down-topic_rel)	0.0192
adjunct(NULL)	0.0128
......	...
obj(up-obj:down-topic_rel)	0.7915
obj(up-obj:up-coord:down-coord:down-obj)	0.1108
......	...
subj(NULL)	0.3903
subj(up-subj:down-topic_rel)	0.2092
......	...

Table 6: Examples of NLD paths

5.2 The Basic Model

The basic algorithm described in Section 4.2 can be used to indiscriminately resolve almost all NLD types for Chinese including locally mediated dependencies with few exceptions (traces with modifier GFs, which accounts for about 1.5% of all NLDs in CTB5.1). Table 7 shows the results of the basic algorithm for trace insertion and antecedent recovery on both stripped CTB trees and parser output trees. For comparison, we implemented the C04 algorithm on our data and evaluated the result. Since the basic algorithm focus on argument traces, results for arguments only are given separately.

Table 7 shows that the C04 algorithm achieves a high precision but as expected a low recall due to its limitation to certain types of NLDs. By contrast, our basic algorithm scored higher recall but lower precision, which is understandable as the C04 algorithm identifies the trace given a known antecedent, whereas our algorithm tries to identify both the trace and antecedent. Compared to trace

	Insertion						Recovery					
	CTB Trees			Parser Output			CTB Trees			Parser Output		
	Prec.	Rec.	F	Prec.	Rec.	F	Prec.	Rec.	F	Prec.	Rec.	F
(Cahill et al., 2004)												
overall	**95.98**	57.86	72.20	**73.00**	40.28	51.91	**90.16**	54.35	67.82	**65.54**	36.16	46.61
args_only	**98.64**	42.03	58.94	**82.69**	30.54	44.60	**86.36**	36.80	51.61	**66.08**	24.40	35.64
Basic Model												
overall	92.44	91.28	**91.85**	63.87	62.15	**63.00**	63.12	62.33	62.72	42.69	41.54	42.10
args_only	89.42	92.95	**91.15**	60.89	63.45	**62.15**	47.92	49.81	48.84	31.41	32.73	32.06
Basic Model with Subject Path Constraint												
overall	92.16	**91.36**	91.76	63.72	**62.20**	62.95	75.96	**75.30**	**75.63**	50.82	**49.61**	**50.21**
args_only	89.04	**93.08**	91.02	60.69	**63.52**	62.07	66.15	**69.15**	**67.62**	42.77	**44.76**	**44.76**

Table 7: Evaluation of trace insertion and antecedent recovery for C04 algorithm, our basic algorithm and basic algorithm with the subject path constraint.

	Insertion						Recovery					
	Basic Model			Hybrid Model			Basic Model			Hybrid Model		
	Prec.	Rec.	F	Prec.	Rec.	F	Prec.	Rec.	F	Prec.	Rec.	F
Overall	92.16	91.36	91.76	**92.86**	**91.45**	**92.15**	75.96	75.30	75.63	**84.92**	**83.64**	**84.28**
SUBJ	92.95	97.81	95.32	94.38	97.81	96.06	66.93	70.42	68.63	81.61	84.57	83.06
OBJ	65.28	64.98	65.13	78.95	55.30	65.04	61.57	61.29	61.43	75.66	53.00	62.33
ADJUNCT	0.0	0.0	0.0	38.24	25.49	30.59	0.0	0.0	0.0	38.24	25.49	30.59
TOPIC	0.0	0.0	0.0	33.33	35.14	34.21	0.0	0.0	0.0	33.33	35.14	34.21
TOPIC_REL	99.85	99.39	99.62	99.85	99.39	99.62	99.85	99.39	99.62	99.85	99.39	99.62
COORD	90.00	100.00	94.74	90.00	100.00	94.74	90.00	100.00	94.74	90.00	100.00	94.74

Table 8: Breakdown of trace insertion and antecedent recovery results on stripped CTB trees for the hybrid model by major grammatical functions.

insertion, the general results for antecedent identification are rather poor. Examining the development data, we found that most recovery errors were due to wrongly treating missing SUBJs as a PRO (using empty NLD paths). Since the subject in Chinese has a very strong tendency to be omitted if it can be inferred from context, the empty NLD path (without any antecedent) has the greatest probability in all resolution paths conditioned on SUBJ, and prevents the SUBJ from finding a proper antecedent in certain cases. To test the effect of the empty path on SUBJ, we weighted non-empty paths for SUBJ so as to suppress the empty path. After testing on the development set, the optimal weight was found to be 1.9. The subject path constraint model shows a dramatic improvement of 12.9% and 8.1% for the overall result of antecedent recovery on CTB trees and parser output trees.

5.3 The Hybrid Fine-Grained Model

As proposed in Section 4.3, we implemented a more fine-grained strategy to capture specific linguistic properties of different NLD types in the CTB. We also combine our basic algorithm (Section 4.2) with (Cahill et al., 2004)'s algorithm in order to resolve the modifier-function traces. The two algorithms may conflict due to (i) inserting the same trace at the same site but related to different antecedents or (ii) resolving the same antecedent to different traces. We keep the traces inserted by the C04 algorithm and abandon those inserted by our algorithm in case of conflict, as the results in Section 5.2 suggest that C04 has a higher precision than ours. Table 8 reports the results of trace insertion and antecedent recovery, respectively, on stripped CTB trees, broken down by major GFs.

The fine-grained hybrid model allows us to recover NLDs with traces with modifier functions and, more importantly it is sensitive to particular linguistic properties of different NLD types. As the hybrid model separates the locally mediated dependencies from other long-distance dependencies, it increases the f-score by 8.7% for antecedent recovery compared with the basic model. Table 9 reports the results of the hybrid model on parser output trees, which shows an increase of 3.6% for antecedent recovery (compared with Table 7).

	Insertion			Recovery		
	Prec.	Rec.	F	Prec.	Rec.	F
overall	64.07	62.37	63.21	54.53	53.08	53.79

Table 9: Evaluation of hybrid model for trace insertion and antecedent recovery on parser output trees.

5.4 Better Training for Parser Output

Our experiments show that although our NLD recovery algorithm performs well on stripped CTB trees, it is sensitive to the noise in parser output trees, with a performance drop of about 30%. This is in contrast to English data, on which (Johnson, 2002) reports a drop of 7-9% moving from treebank trees to parser output trees. No doubt this is partially due to the poor performance of the parser on Chinese data. It is widely accepted that parsing Chinese is more difficult than parsing other more configurational or richer morphological languages, such as English.[7] Our NLD recovery algorithm runs on automatically generated LFG f-structures. The f-structure annotation algorithm is highly tailored to the CTB bracketing scheme (using configurational, categorial and functional tag information), and suffers considerably from errors produced by the parser. Table 4 shows that performance of the f-structure annotation decreases sharply (about 22%) for the parser output trees and this contributes to the eventual trace insertion and antecedent recovery performance drop.

Since the f-structures automatically generated from parser output trees are substantially different from those generated from the original CTB trees, our method to obtain the NLD resolution training data suffers from a serious drawback: the training data come from perfect CTB trees, whereas test data are derived from imperfect parser output trees. This constitutes a serious drawback for machine learning based approaches, such as ours: ideally, instances seen during training should be similar to unseen test data. To make training examples more similar to test instances, we reparse the training set to obtain better training data. To avoid running the parser on the training data, we carried out 10-fold-cross training, dividing the training data into 10 parts and parsing each part in turn with the parser trained on the remaining 9 parts. The reparsed training data are more similar to the test data than the original perfect CTB trees. We then converted both the reparsed training data and the original CTB trees into f-structures, and by comparing with the f-structures generated from the original CTB trees, we recovered the empty nodes and coindexation on the f-structures generated from the reparsed training data. We used parser output based f-structures to train our NLD recovery model and recovered NLDs for parser output trees from the test data. Table 10 presents the results for trace insertion and antecedent recovery on parser output trees using the improved training method, which shows a clear increase in precision and almost the same recall over the normal training (Table 9).

	Insertion			Recovery		
	Prec.	Rec.	F	Prec.	Rec.	F
overall	67.29	62.33	64.71	56.88	52.69	54.71

Table 10: Evaluation of hybrid model for trace insertion and antecedent recovery on parser output trees with better training.

6 Conclusion

We have presented an algorithm for recovering non-local dependencies for Chinese. Our method revises and considerably extends the approach of (Cahill et al., 2004) originally designed for English, and, to the best of our knowledge, is the first NLD recovery algorithm for Chinese. The evaluation shows that our algorithm considerably outperforms (Cahill et al., 2004)'s with respect to Chinese data.

In future work, we will refine and extend the conditioning features in our models to discriminate subcat frames and explore the possibilities to use the Chinese Propbank and Hownet to supplement our automatically acquired subcat frames. We will investigate ways of closing the gap between the performance of gold-standard and parer output trees, including improving parsing result for Chinese. We also plan to adapt other NLD recovery methods (Jijkoun and Rijke, 2004; Schmid, 2006) to Chinese and compare them with the current results.

Acknowledgements

This research is funded by Science Foundation Ireland grant 04/IN/I527.

[7](Bikel, 2004) reports 89% f-score for English parsing of Penn-II treebank data and 79% f-score for Chinese parsing on CTB version 3.

References

Aoife Cahill, Michael Burke, Ruth O'Donovan, Josef van Genabith and Andy Way. 2004. Long-Distance Dependency Resolution in Automatically Acquired Wide-Coverage PCFG-Based LFG Approximations. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 320-327. Barcelona, Spain.

Daniel M. Bikel. 2004. On the Parameter Space of Generative Lexicalized Statistical Parsing Models. *Ph.D. thesis*, Department of Computer & Information Science, University of Pennsylvania. Philadelphia, PA.

Derrick Higgins. 2003. A machine-learning approach to the identification of WH gaps. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, 99-102. Budapest, Hungary.

Eugene Charniak. 2000. A Maximum-Entropy-Inspired Parser. In *Proceedings of the 1st Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 132-139. Seattle, WA.

Helmut Schmid. 2006. Trace Prediction and Recovery With Unlexicalized PCFGs and Slash Features. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 177-184. Sydney, Australia.

Joakim Nivre and Jens Nilsson. 2005. Pseudo-Projective Dependency Parsing. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 99-106. Ann Arbor, Michigan.

Mark Johnson. 2002. A Simple Pattern-Matching Algorithm for Recovering Empty Nodes and Their Antecedents. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 136-143. Philadelphia, PA.

Michael Burke, Olivia Lam, Rowena Chan, Aoife Cahill, Ruth O'Donovan, Adams Bodomo, Josef van Genabith and Andy Way. 2004. Treebank-Based Acquisition of a Chinese Lexical-Functional Grammar. In *Proceedings of the 18th Pacific Asia Conference on Language, Information and Computation*, pages 161-172, Tokyo, Japan.

Michael Collins. 1999. Head-Driven Statistical Models for Natural Language Parsing. *Ph.D. thesis*, Department of Computer & Information Science, University of Pennsylvania. Philadelphia, PA.

Nianwen Xue, Fu-Dong Chiou, and Martha Palmer. 2002. Building a Large-Scale Annotated Chinese Corpus. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1100-1106. Taipei, Taiwan.

Péter Dienes and Amit Dubey. 2003a. Deep syntactic processing by combining shallow methods. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 431-438. Sapporo, Japan.

Péter Dienes and Amit Dubey. 2003b. Antecedent Recovery: Experiments with a Trace Tagger. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 33-40. Sapporo, Japan.

Richard Campbell. 2004. Using Linguistic Principles to Recover Empty Categories. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 645-652. Barcelona, Spain.

Roger Levy and Christopher D. Manning. 2004. Deep Dependencies from Context-Free Statistical Parsers: Correcting the Surface Dependency Approximation. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 327-334. Barcelona, Spain.

Ronald M. Kaplan and Joan Bresnan. 1982. Lexical Functional Grammar: a Formal System for Grammatical Representation. *The Mental Representation of Grammatical Relations*, pages 173-282. MIT Press, Cambridge, MA.

Ryan Gabbard, Seth Kulick, and Mitch Marcus. 2006. Fully Parsing the Penn Treebank In *Proceedings of the Human Language Technology Conference / North American Chapter of the Association of Computational Linguistics*, pages 184-191. New York, USA.

Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell III and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 271-278. Philadelphia, PA.

Stephen Clark and Julia Hockenmaier. 2002. Building Deep Dependency Structures with a Wide-CoverageCCG Parser. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 327-334. Philadelphia, PA.

Valentin Jijkoun. 2003. Finding Non-Local Dependencies: Beyond Pattern Matching. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 37-43. Sapporo, Japan.

Valentin Jijkoun and Maarten de Rijke. 2004. Enriching the Output of a Parser Using Memory-Based Learning. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 311-318. Barcelona, Spain.

Yusuke Miyao, Takashi Ninomiya, and Jun'ichi Tsujii. 2003. Probabilistic Modeling of Argument Structures Including Non-Local Dependencies. In *Proceedings of the 2003 Conference on Recent Advances in Natural Language Processing*, pages 285-291. Philadelphia, PA.