Treebank-Based Acquisition of LFG Resources for Chinese

Yuqing Guo[1], Josef van Genabith[1,2] and Haifeng Wang[3]
[1]NCLT, School of Computing, Dublin City University
[2]IBM Center for Advanced Studies, Dublin, Ireland
[3]Toshiba (China) Research and Development Center, Beijing, China

Proceedings of the LFG07 Conference

Miriam Butt and Tracy Holloway King (Editors)

2007

CSLI Publications

http://csli-publications.stanford.edu/

Abstract

This paper presents a method to automatically acquire wide-coverage, robust, probabilistic Lexical-Functional Grammar resources for Chinese from the Penn Chinese Treebank (CTB). Our starting point is the earlier, proof-of-concept work of (Burke et al., 2004) on automatic f-structure annotation, LFG grammar acquisition and parsing for Chinese using the CTB version 2 (CTB2). We substantially extend and improve on this earlier research as regards coverage, robustness, quality and fine-grainedness of the resulting LFG resources. We achieve this through (i) improved LFG analyses for a number of core Chinese phenomena; (ii) a new automatic f-structure annotation architecture which involves an intermediate dependency representation; (iii) scaling the approach from 4.1K trees in CTB2 to 18.8K trees in CTB version 5.1 (CTB5.1) and (iv) developing a novel treebank-based approach to recovering non-local dependencies (NLDs) for Chinese parser output. Against a new 200-sentence good standard of manually constructed f-structures, the method achieves 96.00% f-score for f-structures automatically generated for the original CTB trees and 80.01% for NLD-recovered f-structures generated for the trees output by Bikel's parser.

1 Introduction

Automatically inducing deep, wide-coverage, constraint-based grammars from existing treebanks avoids much of time and cost involved in manually creating such resources. A number of papers (van Genabith et al., 1999; Sadler et al., 2000; Frank, 2000; Cahill et al., 2002) have developed methods for automatically annotating treebank (phrase structure or c-structure) trees with LFG f-structure information to build f-structure corpora to acquire LFG grammar resources.

In LFG, c-structure and f-structure are independent levels of representation which are related in terms of a correspondence function projection ϕ (Kaplan, 1995). In the conventional interpretation, the ϕ-correspondence between c- and f-structure is defined implicitly in terms of functional annotations on c-structure nodes, from which an f-structure can be computed by a constraint solver.

In one type of treebank-based LFG grammar acquisition approaches, referred to as "annotation-based grammar acquisition", functional schemata are annotated either manually on the entire CFG rules automatically extracted from the treebank (van Genabith et al., 1999); or on a smaller number of hand-crafted regular expression-based templates representing partial and underspecified CFG rules (Sadler et al., 2000) which are applied to automatically annotate the CFG rules extracted from treebank trees; or, using an annotation algorithm traversing treebank trees, applying annotations to each node of a local c-structure subtree in a left/right context partitioned by the head node (Cahill et al., 2002).

An alternative grammar acquisition architecture for LFG, referred to as "conversion-based grammar acquisition", directly induces an f-structure from a c-structure tree, without intermediate functional schemata annotations on c-structure trees. An algorithm building on this architecture was developed in (Frank, 2000) by directly

rewriting partial c-structure fragments into corresponding partial f-structures, using a rewriting system originally developed for transfer-based Machine Translation. As opposed to the CFG rule- and annotation-based architecture in which annotation principles are by and large restricted to local trees of depth one, this approach naturally generalises to non-local trees.

One of the challenges in both the annotation- and more direct conversion-based architectures is to keep the number of f-structure annotation/conversion rules which encode linguistic principles to a minimum, as their creation involves manual effort. Another challenge is to find automatic f-structure annotation/conversion architectures that generalise to different languages and treebank encodings.

A common characteristic of the work cited above is that all the methods are applied to English treebanks (Penn-II, Susanne and AP treebank) from which LFG resources are acquired for English. An initial attempt to extend the treebank- and annotation-based LFG acquisition methodology to Chinese data was carried out by (Burke et al., 2004), which applied a version of (Cahill et al., 2004)'s algorithm adapted to Chinese via the Penn Chinese Treebank version 2 (LDC2001T11) and was evaluated against a small set of 50 manually constructed gold-standard f-structures. The experiments were proof-of-concept and somewhat limited with respect to (i) the coverage of Chinese linguistic phenomena; (ii) the quality of the f-structures produced; (iii) parser output producing only 'proto' f-structures with non-local dependencies unresolved; (vi) the size of the treebank and gold standard.

In the present paper, we address these concerns and present a new f-structure annotation architecture and a new annotation algorithm for Chinese, which:

- combines aspects of both the annotation-based and conversion-based architectures described above.
- generates proper f-structures rather than proto-f-structures by resolving NLDs for parser output.
- scales up to the full Penn Chinese Treebank version 5.1 (LDC2005T01U01), whose size is more than 4 times of that of CTB2.
- is evaluated on a new extended set of Chinese gold-standard f-structures for 200 sentences.[1]

2 Automatic F-Structure Annotation of CTB5.1

2.1 Chinese LFG

Research on LFG has provided analyses for a considerable number of linguistic phenomena in Indo-European, Asian, African and Native American and Australian languages. However, to date, there has been no standard LFG account for many of the core phenomena of Chinese, a language drastically different from English, German, French and other Indo-European languages, which are often the focus of

[1] Developed jointly with PARC.

attention. Chinese has very distinctive linguistic properties, including: (i) very little inflectional morphology encoding tense, number, gender etc., resulting in the almost complete absence of agreement phenomena familiar from European languages; (ii) lack of case markers, complementisers etc., which often causes syntactic and semantic ambiguity; (iii) the tendency towards omission of constituents on condition that they can be inferred from the context, which includes not only subject and object arguments, but also predicates and other heads of phrases, in some cases.

Though the main purpose of this paper is to address the technical issue of automatically inducing f-structures from the Penn Chinese treebank, an LFG account for various phenomena and constructions in Chinese is a prerequisite. To give a flavour of what the Chinese LFG likes look, we illustrate the c-structure trees represented in the CTB and our analyses with the corresponding f-structures for a number of core linguistic phenomena characteristic of Chinese below.

Classifiers are common in Chinese (and some other Asian languages) in that they cooccur with numerals or demonstrative pronouns to count things or persons (nouns) or indicate the frequency of actions (verbs). To provide a unified interpretation of classifiers, we treat a classifier as a grammatical function modifying the head noun (or verb) rather than e.g. as a feature attached to the determiner or head noun/verb, for the following reasons:

- classifiers have content meaning: standard classifiers such as “米/meter”, “公斤/kilogram”, “瓶/bottle” relate to distance, weight, volume, etc. and individual classifiers indicate prominent features of the noun they modify, for example “把/BA” which is derived from “handle” is used as a classifier for objects with a handle, as in (1).

  (1) 一 把 椅子
  one CLS chair
  ‘one chair’

- classifiers can function as the head within a phrase, as in (2).

  (2) 打 三 下
  hit three CLS
  ‘hit three times’

- classifiers can be modified by adjectives, as in (3).

  (3) 一 大 碗 饭
  one big bowl/CLS rice
  ‘a big bowl of rice’

Figure 1 illustrates the CTB representation of a classifier and the corresponding schematic f-structure. A noticeable difference is that the determiner (DT) takes a quantifier phrase (QP) as its complement in the CTB constituent-tree, whereas in our f-structure the determiner and quantifier are parallel functions both specifying the head noun predicate.

(4) 这 五 个 学生
these five CLS student
'these five students'

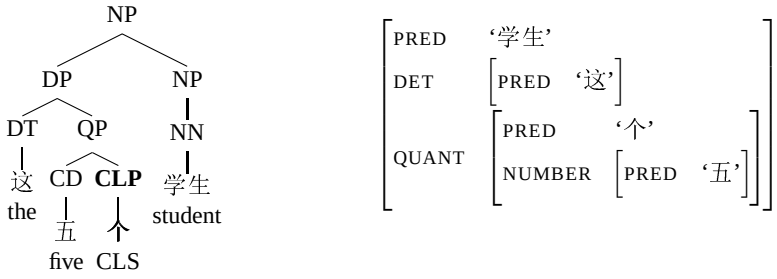
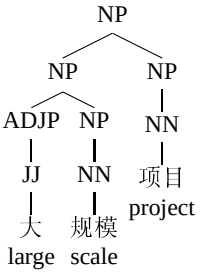


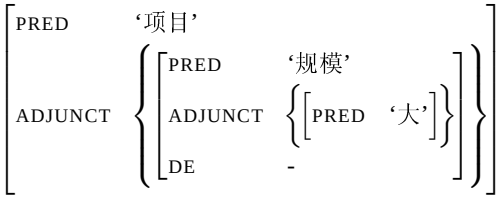
Figure 1: The CTB tree and our f-structure analysis of classifier

DE Phrases are formed by the function word "的/DE" attached to various categories, such as possessive phrases, noun phrases, adjective phrases or relative clauses. DE has no content other than marking the preceding phrase as a modifier of NP. Different from the original f-structure annotation algorithm and the 50-sentence gold-standard f-structures developed in (Burke et al., 2004), we choose the content word rather than DE as head of the modifier, because all the other words in the modifier phrase will depend on the head, and moreover DE has no content thus may be omitted in examples such as (5a). Therefore, in our analysis we treat DE as an optional feature attached to the modifier as exemplified in Figure 2. What is noticeable here is that the grammatical function of the DE-phrase in (5b) is an attributive modifier (ADJUNCT) while in (6) it is a possessor (POSS), even though the constituent structures are the same for both, due to the absence of any case marking. The difference is in fact lexical and due to the head word of the adjunct which is a common noun (NN) in (5), and the head word of the possessor which is a proper noun (NR) in (6).

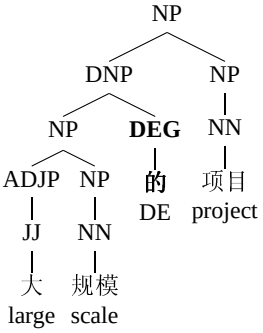
BEI-Constructions are commonly considered approximately equivalent to passive voice in English. However we do not treat "被/BEI" as just a passive voice feature, in that it also introduces the logic subject in long-BEI constructions as in (7), similar to the preposition "by" in the English passive construction. Furthermore, we do not analyse it as a subject marker, as short-BEI constructions as in (8) will be subjectless, where BEI marks nothing. And rather than treating it as a preposition, though the analysis can be argued from a theoretical point of view, it does not always indicate passive voice, as in (9), where the embedded verb is intransitive. In line with (Her, 1991), we treat BEI as a verb. The advantage of this analysis is that it provides a unified account for embedded verbs, where verbs in BEI sentences have the same subcategorisation frames as those in their BEI-less corresponding sentences. (Her, 1991) treats BEI as a pivotal construction, where BEI requires an object and an non-finite VP complement. However, this

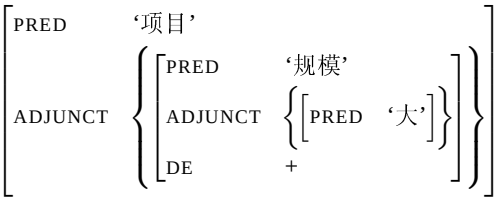
(5) a. 大　规模 项目
large scale project



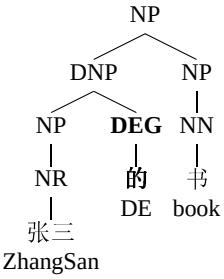


b. 大　规模 的 项目
large scale DE project
‘a large-scale project’





(6) 张三　　的 书
ZhangSan DE book
‘ZhangSan’s book’



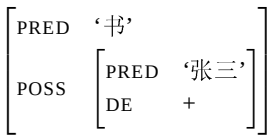
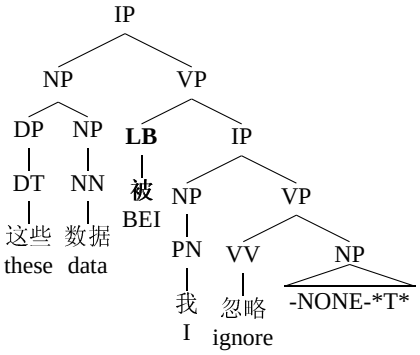


Figure 2: The CTB tree and our f-structure analysis of DE-phrase

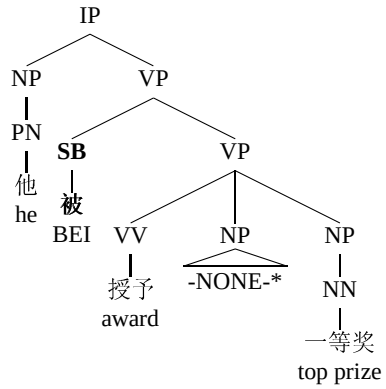
is somewhat different from the CTB representation, where BEI takes a sentential complement. Both constructions are acceptable in Chinese without the presence of a complementiser. For practical purposes, we accept the tree representation in CTB and hence BEI requires a closed complement (COMP) in our f-structure, as exemplified in Figure 3.

(7) 这些 数据 被 我 忽略
these data BEI I ignore
'These data was ignored by me.'



$$\begin{bmatrix} \text{PRED} & \text{'被⟨SUBJ, COMP⟩'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'数据'} \\ \text{DET} & \begin{bmatrix} \text{PRED} & \text{'这些'} \end{bmatrix} \end{bmatrix} \boxed{1} \\ \text{COMP} & \begin{bmatrix} \text{PRED} & \text{'忽略⟨SUBJ, OBJ⟩'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'我'} \end{bmatrix} \\ \text{OBJ} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

(8) 他 被 授予 一等奖
he BEI award the top prize
'He was awarded the top prize.'



$$\begin{bmatrix} \text{PRED} & \text{'被⟨SUBJ, COMP⟩'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'他'} \end{bmatrix} \boxed{1} \\ \text{COMP} & \begin{bmatrix} \text{PRED} & \text{'授予⟨SUBJ, OBJ, OBL\_TH⟩'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'pro'} \end{bmatrix} \\ \text{OBJ} & \boxed{1} \\ \text{OBL\_TH} & \begin{bmatrix} \text{PRED} & \text{'一等奖'} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 3: The CTB tree and our f-structure analysis of BEI-construction

(9) 猫 被 老鼠 跑 了
cat BEI mouse escape ASP
'The cat let the mouse escape.'

2.2 A New F-Structure Annotation Algorithm for CTB

The f-structure annotation method developed in (Cahill et al., 2002; Burke et al., 2004) builds on CFG rule- and annotation-based architecture. By and large the algorithm works on local treebank subtrees of depth one (equivalent to a CFG rule)[2]. In order to annotate the nodes in the tree, the algorithm partitions each sequence of daughters in the local subtree into three sections: left context, head and right

[2]Though it also uses some non-local information.

context. Configurational information (left or right position regarding to the head), category of mother and daughter nodes and Penn treebank functional labels (if they exist) on daughter nodes are exploited to annotate nodes with f-structure functional equations. The annotation principles for Chinese in (Burke et al., 2004) are fairly coarse-grained. However configurational and categorial information from local trees of depth one only is not always sufficient to determine the appropriate grammatical function (GF), as for example for DE-phrases (Figure 2). This means disambiguation of GFs for Chinese may require access to lexical information (common or proper noun in Figure 2) and more extensive contextual information beyond the local configurational and categorial structure.

In (Cahill et al., 2002; Burke et al., 2004), for each tree, the f-structure equations are collected after annotation and passed on to a constraint solver which produces an f-structure for the tree. Unfortunately, as explained in (Cahill et al., 2002), the constraint solver's capability is limited: it can handle equality constraints, disjunction and simple set-valued feature constraints. However, it (i) fails to generate an f-structure (either complete or partial) in case of clashes between the automatically annotated features; and (ii) does not provide subsumption constraints to distribute distributive features into coordinate f-structures.

In order to avoid the limitations of the constraint solver, and in order to exploit more information for function annotation from a larger context rather than within the local tree, instead of indirectly generating the f-structure via functional equations annotated to c-structure trees, we adopt the alternative approach which transduces the treebank tree into f-structure via an intermediate dependency structure, directly constructed from the original c-structure tree, as shown in Figures 4 and 5.

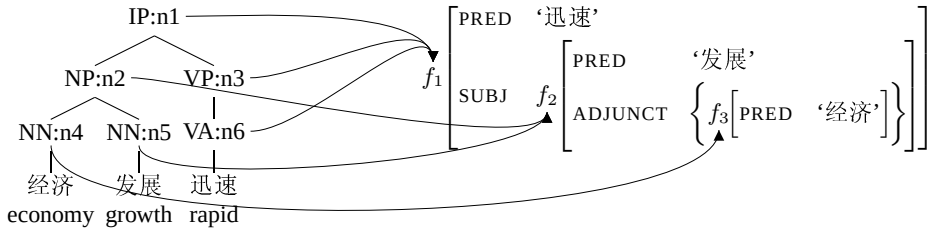
The basic idea is that the $\uparrow{=}\downarrow$ (or the equivalent $\phi(n_i){=}\phi(n_j)$ equations in Figure 4) head projections in the classical LFG projection architecture allow us to collapse a c-structure tree into an intermediate, unlabelled dependency structure as in Figure 5. The intermediate unlabelled dependency structure is somewhat more abstract and normalised (compared to the original c-structure tree) and is used as input to an f-structure annotation algorithm, which is simpler and more general than the conventional f-structure algorithms (Cahill et al., 2002; Burke et al., 2004) directly operating on the original, more complex and varied c-structure trees.

The new f-structure annotation architecture is illustrated in Figure 5, and includes two major steps:

I. We first extract all predicates from the (local) c-structure tree, using head-finding rules similar to that used in (Collins, 1999), adapted to Chinese data and CTB5.1. Collapsing head-branches along the head-projection lines, the c-structure configuration is projected to an intermediate unlabelled dependency structure, augmented with CFG category and order information inherited from the c-structure.

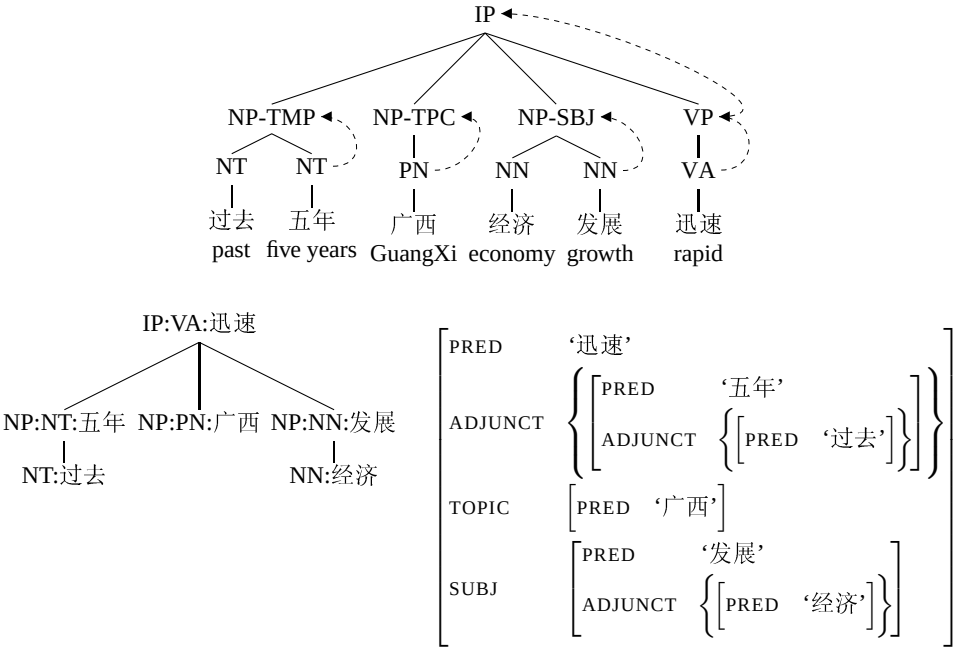
II. Second, we use high-level annotation principles exploiting configurational, categorial, functional as well as lexical information from the intermediate

unlabelled dependency structure to annotate grammatical function and other f-structure information (to create a labelled dependency structure, i.e. an LFG f-structure).



ϕ-correspondence:	f-structure
ϕ(n1)=ϕ(n3)=ϕ(n6)=f_1	(f_1 PRED)=‘迅速’ (f_1 SUBJ)=f_2
ϕ(n2)=ϕ(n5)=f_2	(f_2 PRED)=‘发展’ (f_2 ADJUNCT)=f_3
ϕ(n4)=f_3	(f_3 PRED)=‘经济’

Figure 4: ϕ-projection from c-structure to f-structure



(I) Predicate Extraction (II) Function Annotation

Figure 5: The new f-structure annotation architecture for CTB

By abstracting away from the ‘redundant’ c-structure nodes in our intermediate dependency representations, the annotation principles can apply to non-local sub-trees. This allow us to disambiguate different GFs in a larger context and resort to

lexical information. As a more abstract dependency-like structure is used to mediate between the c- & f-structure, the algorithm always generates an f-structure, and there are no clashing functional equations causing the constraint solver to fail. Moreover, the intermediate dependency structure can easily handle distribution into coordinate structures by moving and duplicating the dependency branch associated with distributive functions. Furthermore, finite approximations of functional uncertainty equations resembling paths of non-local dependencies also can be computed on the intermediate dependency structure for the purpose of NLD recovery (this will be presented in section 3). Finally, in order to conform to the coherence condition and to produce a single connected f-structure for every CTB tree, a post-processing step is carried out to check duplications and to catch and add missing annotations.

Our new annotation algorithm is somewhat similar in spirit to the conversion approach developed in (Frank, 2000), However in (Frank, 2000)'s algorithm the mapping of c-structure to f-structure is carried out in one step using a tree/graph rewriting system. Our method enforces a clear separation between the intermediate unlabelled dependency structure (predicate identification) and function annotation. Predicate identification maps c-structure into an unlabelled dependency representation, and is thus designed particularly for a specific type of treebank encoding and data-structures. By contrast, function annotation is accomplished on the dependency representation which is much more compact and normalised than the original c-structure representation, hence the function annotation rules are more simple and the architecture minimises the dependency of the annotation rules on the particulars of the particular treebank encoding.

2.3 Experimental Evaluation

Similar to (Cahill et al., 2002; Burke et al., 2004), our new annotation algorithm is evaluated both quantitatively and qualitatively.

We apply the f-structure annotation algorithm to the whole CTB5.1 with 18,804 sentences. Unlike the CFG- and annotation-based predecessors (Cahill et al., 2002; Burke et al., 2004), the new algorithm guarantees that 100% of the treebank trees receive a single, connected f-structure.

For the purpose of qualitative evaluation, we selected 200 sentences from CTB5.1 for which the f-structures are automatically produced by our annotation algorithm, and then manually corrected to construct a gold-standard set in line with our Chinese LFG analyses represented in Section 2.1. Annotation quality is measured in terms of predicate-argument-adjunct (or dependency) relations. The relations are represented as triples $relation(predicate, argument/adjunct)$, following (Crouch et al., 2002). The f-structure annotation algorithm is applied to two different sets of test data: (i) the original CTB trees, and (ii) trees output by Bikel's parser (Bikel and Chiang, 2000) trained on 80% of the CTB5.1 trees, exclusive of the 200 gold-standard sentences. Table 1 reports the results against the new 200-sentence set of gold-standard f-structures.

	CTB Trees			Parser Output Trees		
	Precision	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**
Preds Only	93.68	94.93	94.30	73.55	65.05	69.04
All GFs	95.25	96.75	96.00	84.00	71.77	77.40

Table 1: Quality of f-structure annotation

Table 1 shows that given high-quality input trees, the new algorithm produces high quality f-structures with f-scores of around 94%-96% for preds-only and all GFs, respectively. The corresponding scores drop by 20%-24% absolute on parser produced trees.

3 Recovery of Chinese Non-Local Dependencies for Parser Output

The drastic drop in the results on parser output trees is mainly due to labelled bracketing parser errors, but also because Bikel's parser (and most state-of-the-art treebank-based broad-coverage probabilistic parsers) does not capture non-local dependencies (or 'movement' phenomena)[3]. As a result, the automatically generated f-structures produced from parser output trees are proto-f-structures, as they only represent purely local dependencies. In this section, we present a post-processing approach to recover NLDs on the automatically generated proto-f-structures.

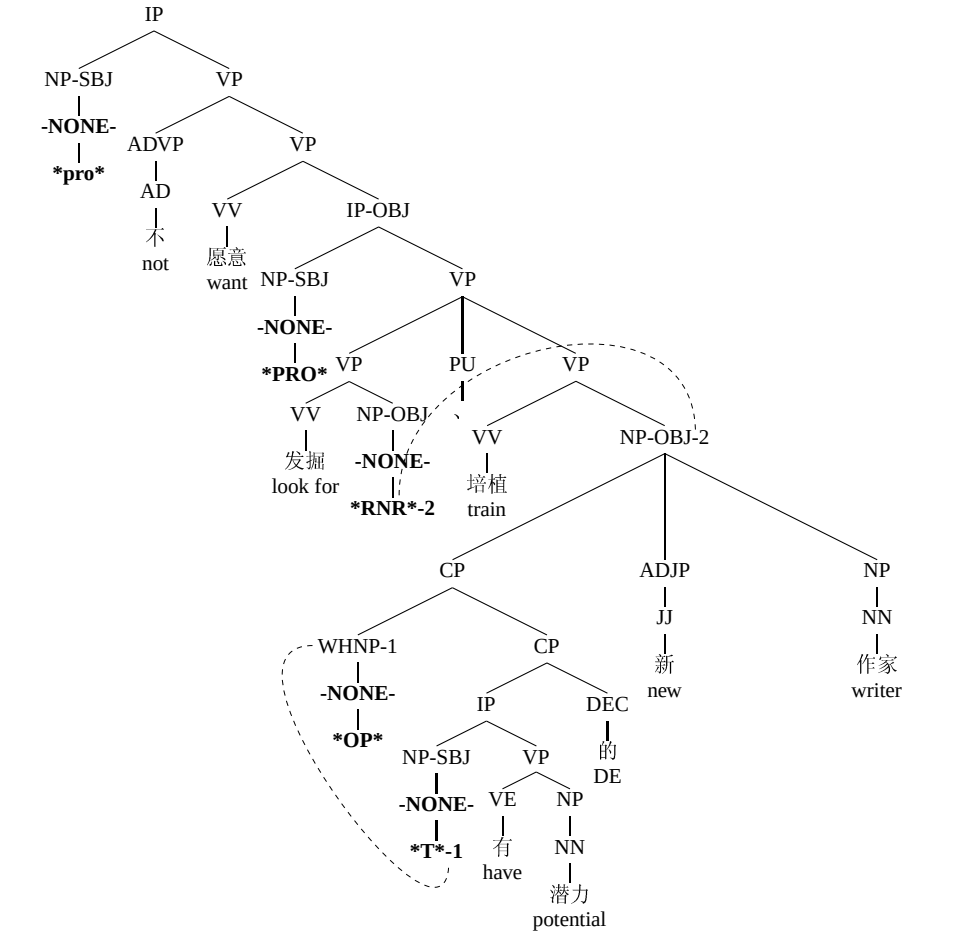
3.1 NLDs in Chinese

Non-local dependencies in CTB are represented in terms of empty categories (ECs) and (for some of them) coindexation with antecedents, as exemplified in Figure 6. Following previous work for English and the CTB annotation scheme (Xue and Xia, 2000), we use the term "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.

Table 2 gives a breakdown of the most frequent types of empty categories and their antecedents. According to their different linguistics properties, we classify these empty nodes into three major types: null relative pronouns, locally mediated dependencies, and long-distance dependencies (LDDs).

Null Relative Pronouns (Table 2, rows 2 and 7) themselves are local dependencies, and thus are not coindexed with an antecedent. But they mediate non-local

[3]The original parser does not produce CTB functional tags either, of which the f-structure annotation algorithm takes advantage (if they are present). To restore the CTB functional tags, we retrained the original parser to allow it to produce CTB functional tags as part its output.



$$
\begin{bmatrix}
\text{PRED} & \text{‘愿意⟨SUBJ, XCOMP⟩’} \\
\text{ADJUNCT} & \begin{bmatrix} \text{PRED} & \text{‘不’} \\ \text{ADJ-TYPE} & \text{neg} \end{bmatrix} \\
\text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{‘pro’} \\ \text{PRO\_TYPE} & \text{pers} \end{bmatrix} \boxed{1} \\
\text{XCOMP} & \begin{bmatrix}
\text{COORD} & \left\{ \begin{bmatrix} \text{PRED} & \text{‘发掘⟨SUBJ, OBJ⟩’} \\ \text{SUBJ} & \boxed{1} \\ \text{OBJ} & \boxed{2} \end{bmatrix} \right. \\
& \left. \begin{bmatrix} \text{PRED} & \text{‘培植⟨SUBJ, OBJ⟩’} \\ \text{SUBJ} & \boxed{1} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{‘作家’} \\ \text{ADJUNCT} & \left\{ \begin{bmatrix} \text{PRED} & \text{‘新’} \end{bmatrix} \right\} \\ \text{ADJUNCT\_REL} & \left\{ \begin{bmatrix} \text{PRED} & \text{‘有⟨SUBJ, OBJ⟩’} \\ \text{TOPIC\_REL} & \begin{bmatrix} \text{PRED} & \text{‘pro’} \\ \text{PRO\_TYPE} & \text{rel} \end{bmatrix} \boxed{3} \\ \text{SUBJ} & \boxed{3} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{‘潜力’} \end{bmatrix} \end{bmatrix} \right\} \end{bmatrix} \boxed{2} \end{bmatrix} \right\} \\
\text{COORD\_FORM} & \text{‘、’}
\end{bmatrix}
\end{bmatrix}
$$

Figure 6: NLDs example of sentence ‘*(People) don’t want to look for and train new writers who have potential.*’: the CTB tree and the corresponding f-structure.

	Antecedent	POS	Label	Count	Description
1	WHNP	NP	*T*	11670	WH trace (e.g. *OP*中国发射***T***的卫星)
2		WHNP	*OP*	11621	Empty relative pronouns (e.g. ***OP***中国发射的卫星)
3		NP	*PRO*	10946	Control constructions (e.g. 这里不许***PRO***抽烟)
4		NP	*pro*	7481	Pro-drop situations (e.g. ***pro***不曾遇到的问题)
5	IP	IP	*T*	575	Topicalisation (e.g. 我们能赢，他说***T***)
6	WHPP	PP	*T*	337	WH trace (e.g. *OP*人口***T***密集地区)
7		WHPP	*OP*	337	Empty relative pronouns (e.g. ***OP***人口密集地区)
8	NP	NP	*	291	Raising & passive constructions (e.g. 我们被排除*****在外)
9	NP	NP	*RNR*	258	Coordinations (e.g. 鼓励***RNR***和支持投资)
10	CLP	CLP	*RNR*	182	Coordinations (e.g. 五***RNR***至十亿元)
11	NP	NP	*T*	93	Topicalisation (e.g. 薪水都用***T***来享乐)

Table 2: The distribution of the most frequent types of empty categories and their antecedents in CTB5.1.

dependencies by functioning as antecedents for the dislocated constituent inside a relative clause.[4]

Locally Mediated Dependencies are non-local in that 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: (Table 2, row 8) raising constructions, and short-bei constructions (passivisation); (row 3) control constructions, which includes two different types: a generic *PRO* with an arbitrary reading (approximately equal 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 differ from locally mediated dependencies, in that the path linking the antecedent and trace might be unbounded. LDDs include the following phenomena:

Wh-traces in relative clauses, where an argument (Table 2, row 1) or adjunct (row 6) 'moves' and is coindexed with the 'extraction' site.

Topicalisation (Table 2, rows 5 and 11) is one of the typical LDDs in English, whereas in Chinese not all topics involve displacement, as shown in example (10).

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

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

[5]However in this case the CTB annotation does not coindex the locus (trace) with its controller (antecedent) as the *PRO* in Figure 6.

Long-Bei construction as described above, taking a sentential complement which possibly involves long-distance dependencies, as in example (11).

(11) 约翰 被 玛丽 派 人 打 了
John BEI Mary send somebody hit ASP
'John was hit by somebody sent by Mary.'

Coordination is divided into two groups: right node raising of an NP phrase which is an argument shared by the coordinate predicates (Table 2, row 9); and the coordination of quantifier phrases (row 10) and verbal phrases as example (12), in which the antecedent and trace are both predicates and possibly take their own arguments or adjuncts.

(12) 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 cases (Table 2, row 4) are prominent in Chinese because subject and object functions are only semantically but not syntactically required. Nevertheless, here 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)[6] context.

3.2 NLD Recovery Algorithm for CTB

Among these NLD types, LDDs cover various linguistic phenomena and are the most difficult to resolve. Inspired by (Cahill et al., 2004), we recover long-distance dependencies at the level of f-structures, using automatically acquired subcategorisation frames and finite approximations of functional uncertainty equations describing LDD paths from the f-structure annotated CTB. (Cahill et al., 2004)'s algorithm only resolves certain LDDs with known types of antecedents (TOPIC, TOPIC_REL and FOCUS). However as illustrated above, except for relative clauses, the antecedents in Chinese LDDs do not systematically correspond to types of grammatical function. Furthermore, more than half of all empty categories are not coindexed with an antecedent due to the high prevalence of pro-drop in Chinese. In order to resolve all Chinese LDDs represented in the CTB, we modify and substantially extend (Cahill et al., 2004)'s algorithm as follows:

1. we extract LDD resolution paths p linking reentrances in f-structures automatically generated for the original CTB trees. To better account for all Chinese LDDs represented in the CTB, we calculate the probability of p conditioned on the GF associated with the trace t (instead of the antecedent

[6] In this case, the 'pro' will be resolved by anaphora resolution in a later stage.

as in Cahill et al. (2004)). The path probability $P(p|t)$ is estimated as Eq. 1 and some examples of LDD paths are listed in Table 3.

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

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 3: Examples of LDD paths

2. we extract the subcat frames s for each verbal form w from the automatically generated f-structures and calculate the probability of s conditioned on w. As Chinese has little inflectional morphology, we augment the word w with syntactic features including the POS of w, the GF of w, so as to disambiguate subcat frames and choose the appropriate one in particular context. The lexical subcat frame probability $P(s|w, w\_feats)$ is estimated as Eq. 2 and some examples of subcat frames are listed in Table 4.

$$P(s|w, w\_feats) = \frac{count(s, w, w\_feats)}{\sum_{i=1}^{n} count(s_i, w, w\_feats)} \quad (2)$$

Word:POS-GF(Subcat Frames)	**Prob.**
有:VE-adj_rel([subj, obj])	0.6769
有:VE-adj_rel([subj, comp])	0.1531
有:VE-adj_rel([subj])	0.0556
.....	...
有:VE-comp([subj, obj])	0.4805
有:VE-comp([subj, comp])	0.2587
.....	...
有:VE-top([subj, comp])	0.4397
有:VE-top([subj, obj])	0.3510
.....	...

Table 4: Examples of subcat frames

3. given the set of subcat frames s for the word w, and the 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 inside-out 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

4. rank all resolution candidates according to the product of subcat frame and LDD path probabilities (Eq. 3).

$$P(s|w, w\_feat) \times \prod_{j=1}^{m} P(p|t_j) \tag{3}$$

As described in Section 3.1, besides LDDs, there are two other types of NLDs in the CTB5.1, and their different linguistic properties may require more fine-grained recovery strategies than the one described so far. Furthermore, as the LDD recovery method described above is triggered by dislocated subcategorisable grammatical functions, cases of LDDs in which the trace is not an argument in the f-structure, e.g. an ADJUNCT or TOPIC in relative clauses or a null PRED in verbal coordination, can not be recovered by the algorithm. In order to recover all NLD types in the CTB5.1, we develop a hybrid methodology. The hybrid method involves four strategies (including the one described so far):

- 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 (Cahill et al., 2004)'s algorithm, which conditions the probability of LDD path on the GF associated of the antecedent rather than the trace, to resolve the wh-trace in relativisation, including ungovernable GFs TOPIC and ADJUNCT.
- Using our modified LDD resolution algorithm to resolve the remaining types.

3.3 Experimental Evaluation

For the experiments on NLD recovery, we use the first 760 articles of CTB5.1, from which 75 double-annotated files (1,046 sentences) are used as test data, 75

files (1,082 sentences) are held out as development data, while the other 610 files (8,256 sentences) are used as training data. Experiments are carried out on two different kinds of input: first on CTB gold standard trees stripped of all empty nodes and coindexation information; and second, on the output trees of Bikel's parser.

We use the triple dependency relation encoding in the evaluation metric for NLD recovery. In the trace insertion evaluation, the trace is represented by the empty category, e.g. OBJ(发掘/look for, NONE) in Figure 6; and in the antecedent recovery evaluation, the trace is realised by the predicate of the antecedent, e.g. OBJ(发掘/look for, 作家/writer).

Table 5 shows the performance of the NLD recovery algorithm against (i) the CTB5.1 test set given the trees stripped of all empty nodes and coindexation information and (ii) output trees by Bikel's parser. Table 6 gives the results of f-structure annotation for parser output after NLD resolution evaluated against the 200-sentence gold standard, which shows 2.3% and 2.6% improvement of pred-only measure and all-GFs measure respectively over the proto-f-structures (Table 1).

	CTB Trees			**Parser Output Trees**		
	Precision	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**
Insertion	92.86	91.45	92.15	67.29	62.33	64.71
Recovery	84.92	83.64	84.28	56.88	52.69	54.71

Table 5: Evaluation of NLD trace insertion and antecedent recovery

+NLD res.	**Precision**	**Recall**	**F-Score**
Preds Only	71.91	70.81	71.36
All GFs	80.41	79.61	80.01

Table 6: Evaluation of proper f-structures from NLD-resolved parser output

4 Conclusions and Future Work

We have reported on a project on inducing wide-coverage LFG approximations for Chinese from the CTB5.1. Our new two-stage annotation architecture provides an interface transducing c-structure trees to f-structures. The method avoids some of the limitations of the CFG rule- and annotation-based method. The more general annotation principles operating on intermediate unlabelled dependency representations allow us to scale the method to the whole Penn Chinese treebank and guarantee that every constituent-tree in the CTB5.1 can derive a complete f-structure. The separation of function annotation from the determination of the unlabelled dependency representations, minimises the dependency of the functional annotation

principles on the particular treebank encoding and data-structures. Our f-structure annotation algorithm is motivated by Chinese, however, in large parts it is less language-dependent than the CFG-rule- and annotation-based methods of (Cahill et al., 2002; Burke et al., 2004). As the method exploits information from a larger context, including non-local trees and lexical information, it may also benefit less configurational languages which exhibit relatively free word order, with morphology rather than phrasal position determining functional roles. Finally, the non-local dependency recovery method captures 'moved' constituents and produces a full-fledged f-structure from parser output.

Areas of current and future research include further extending the gold-standard and examining more kinds of constructions and linguistic phenomena particular in Chinese. We will also investigate ways of closing the gap between the performance of CTB trees and parer output trees, including improving parsing result for Chinese.

Acknowledgements

We gratefully acknowledge support from Science Foundation Ireland grant 04/IN/I527 for the research reported in this paper.

References

Bikel, D.M. and Chiang, D. 2000. Two Statistical Parsing Models Applied to the Chinese Treebank. In *Proceedings of the Second Chinese Language Processing Workshop*, pages 1–6, Hong Kong, China.

Burke, M., Lam, O., Chan, R., Cahill, A., O'Donovan, R., Bodomo, A., van Genabith, J. and Way, A. 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.

Cahill, A., Burke, M., O'Donovan, R., van Genabith, J. and Way, A. 2004. Long-Distance Dependency Resolution in Automatically Acquired Wide-Coverage PCFG-Based LFG Approximations. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, pages 319–326, Barcelona, Spain.

Cahill, A., McCarthy, M., van Genabith, J. and Way, A. 2002. Automatic Annotation of the Penn Treebank with LFG F-Structure Information. In *Proceedings of the LREC Workshop on Linguistic Knowledge Acquisition and Representation: Bootstrapping Annotated Language Data*, pages 8–15, Las Palmas, Canary Islands, Spain.

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

Crouch, R., Kaplan, R.M., King, T.H. and Riezler, S. 2002. A comparison of evaluation metrics for a broad coverage parser. In *Proceedings of the LREC Workshop: Beyond PARSEVAL – Towards Improved Evaluation Measures for Parsing Systems*, pages 67–74, Las Palmas, Canary Islands, Spain.

Frank, A. 2000. Automatic F-Structure Annotation of Treebank Trees. In Miriam Butt and Tracy Holloway King (eds.), *Proceedings of the Fifth International Conference on Lexical Functional Grammar*, pages 226–243, Berkeley, CA: CSLI Publications.

Her, O.S. 1991. *Grammatical Functions and Verb Subcategorization in Mandarin Chinese*. Taipei: Crane Publishing.

Kaplan, R.M. 1995. The formal architecture of lexical-functional grammar. In John T. Maxwell III Mary Dalrymple, Ronald M. Kaplan and Annie Zaenen (eds.), *Formal Issues in Lexical-Functional Grammar*, pages 7–27, Standford, USA: CSLI Publications.

Sadler, L., van Genabith, J. and Way, A. 2000. Automatic F-Structure Annotation from the AP Treebank. In Miriam Butt and Tracy Holloway King (eds.), *Proceedings of the Fifth International Conference on Lexical Functional Grammar*, pages 226–243, Berkeley, CA: CSLI Publications.

van Genabith, J., Sadler, L. and Way, A. 1999. Semi-automatic Generation of F-Structures from Treebanks. In Miriam Butt and Tracy King (eds.), *Proceedings of the Fourth International Conference on Lexical Functional Grammar*, Manchester, UK.

Xue, N.W. and Xia, F. 2000. The Bracketing Guidlines for the Penn Chinese Treebank (3.0). Technical Report 00-08, Institute for Research in Cognitive Science, University of Pennsylvania.