Treebank-Based Acquisition of a Chinese Lexical-Functional Grammar

Michael BURKE
[†]National Centre for Language Technology,
School of Computing, Dublin City University
and [‡]Centre for Advanced Studies, IBM,
Dublin, Ireland.
mburke@computing.dcu.ie

Olivia LAM
[§]Department of Linguistics,
The University of Hong Kong,
Pokfulam, Hong Kong.
olivia@hku.hk

Aoife CAHILL[†]
acahill@computing.dcu.ie

Rowena CHAN[§]
rowenac@graduate.hku.hk

Ruth O'DONOVAN[†]
rodonovan@computing.dcu.ie

Adams BODOMO[§]
abbodomo@hku.hk

Josef van GENABITH[†‡]
josef@computing.dcu.ie

Andy WAY[†‡]
away@computing.dcu.ie

Abstract

Scaling wide-coverage, constraint-based grammars such as Lexical-Functional Grammars (LFG) (Kaplan and Bresnan, 1982; Bresnan, 2001) or Head-Driven Phrase Structure Grammars (HPSG) (Pollard and Sag, 1994) from fragments to naturally occurring unrestricted text is knowledge-intensive, time-consuming and (often prohibitively) expensive. A number of researchers have recently presented methods to automatically acquire wide-coverage, probabilistic constraint-based grammatical resources from treebanks (Cahill et al., 2002, Cahill et al., 2003; Cahill et al., 2004; Miyao et al., 2003; Miyao et al., 2004; Hockenmaier and Steedman, 2002; Hockenmaier, 2003), addressing the knowledge acquisition bottleneck in constraint-based grammar development. Research to date has concentrated on English and German. In this paper we report on an experiment to induce wide-coverage, probabilistic LFG grammatical and lexical resources for Chinese from the Penn Chinese Treebank (CTB) (Xue et al., 2002) based on an automatic f-structure annotation algorithm. Currently 96.751% of the CTB trees receive a single, covering and connected f-structure, 0.112% do not receive an f-structure due to feature clashes, while 3.137% are associated with multiple f-structure fragments. From the f-structure-annotated CTB we extract a total of 12975 lexical entries with 20 distinct subcategorisation frame types. Of these 3436 are verbal entries with a total of 11 different frame types. We extract a number of PCFG-based LFG approximations. Currently our best automatically induced grammars achieve an f-score of 81.57% against the trees in unseen articles 301-325; 86.06% f-score (all grammatical functions) and 73.98% (preds-only) against the dependencies derived from the f-structures automatically generated for the original trees in 301-325 and 82.79% (all grammatical functions) and 67.74% (preds-only) against the dependencies derived from the manually annotated gold-standard f-structures for 50 trees randomly selected from articles 301-325.

1 Introduction

Scaling wide-coverage, constraint-based grammars such as Lexical-Functional Grammars (LFG) (Kaplan and Bresnan, 1982; Bresnan, 2001) or Head-Driven Phrase Structure Grammars (HPSG) (Pollard and Sag, 1994) from fragments to naturally occurring unrestricted text is knowledge-intensive, time-consuming and (often prohibitively) expensive, constituting an instance of the knowledge acquisition bottleneck familiar from other traditional rule-based approaches in AI and NLP.

Starting with Charniak (1996), many researchers have explored automatic grammar acquisition methods where grammatical information is induced from treebanks. This approach incurs low

development cost and produces wide-coverage, robust, state-of-art resources. However, (with few exceptions) the grammars induced are mostly "shallow", i.e. without the deep syntactic (dependency) or semantic information captured by deep, constraint-based grammar formalisms such as LFG or HPSG.

A recent body of research had extended the basic paradigm of automatic PCFG acquisition from treebanks to the extraction of deep, wide-coverage, constraint-based grammars and lexical resources such as LFG (Cahill et al., 2002; Cahill et al., 2003; Cahill et al., 2004; O'Donovan et al., 2004), HPSG (Miyao et al., 2003; Miyao et al., 2004) and CCG (Hockenmaier and Steedman, 2002; Hockenmaier, 2003). Cahill et al. have developed a methodology for the automatic f-structure annotation of treebanks from which LFG grammars and lexical resources are extracted. To date this research has been applied to the Penn-II treebank (Marcus et al., 1994) for English and the TIGER treebank (Brants et al., 2002) for German. In this paper, we report on an experiment to extend this research to a new language—Mandarin Chinese—via the Penn Chinese Treebank (CTB) (Xue et al., 2002).

In Section 2 we first give a brief review of Lexical-Functional Grammar. Section 3 provides a short description of the CTB (Xue et al., 2002). We present an automatic f-structure annotation algorithm for the CTB. The algorithm generates proto-f-structures (Cahill et al., 2002). Proto-f-structures capture basic but possibly incomplete predicate-argument-adjunct structure as they do not yet resolve long-distance dependencies. Section 4 outlines the architecture underlying the automatic f-structure annotation algorithm and how it was applied to the CTB. Section 5 provides an evaluation of the f-structures produced by the annotation algorithm against a gold-standard of f-structures for 50 randomly selected trees from articles 301-325 CTB. Section 6 details the process of extracting lexical resources from the f-structure-annotated CTB. The extraction of PCFG- (Probabilistic Context Free Phrase Structure Grammar-) based LFG approximations from the f-structure-annotated CTB is presented and evaluated in Section 7. Conclusions and an outline of ongoing and future work are provided in Section 8.

2 Lexical-Functional Grammar

Lexical-Functional Grammar (LFG) (Kaplan and Bresnan, 1982; Bresnan, 2001) is an early member of the family of constraint-based grammar formalisms (FUG, PATR-II, GPSG, HPSG, etc.). It enjoys continued popularity in theoretical and computational linguistics and natural language processing applications and research. At its most basic, an LFG involves two levels of representation: c-structure (constituent structure) and f-structure (functional structure). C-structure represents surface grammatical configurations such as word order and the grouping of linguistic units into larger phrases. The c-structure component of an LFG is represented by a CF-PSG (context-free phrase structure grammar). F-structure represents abstract syntactic functions such as sub(ject), obj(ect), pred(icate), sentential comp(lement), open xcomp(lement), adj(unct), app(osition) etc. in terms of recursive attribute-value structure representations approximating to basic predicate-argument-adjunct or dependency structure. These syntactic representations abstract away from the particulars of surface configuration. The motivation is that while languages differ with respect to surface representation they may still encode the same (or very similar) abstract syntactic functions (or predicate-argument structure).

3 Penn Chinese Treebank version 3.0 (CTB)

The Penn Chinese Treebank (CTB) version 3.0 (Xue et al., 2002) consists of 4185 sentences of Xinhua newswire text in Mandarin Chinese (with 99,529 words – about a tenth of the Penn-II treebank (Marcus et al., 1994)) in 325 articles. Chinese is subject pro-drop and exhibits little morphological marking. The CTB assumes that Mandarin Chinese is strictly configurational. The CTB annotation scheme involves 33 POS-tags, 17 phrasal tags, 6 verb compound tags, 7 empty category tags and 26 functional tags. The CTB functional tags (Tag) can be attached to phrasal tags (Cat) to form Cat-Tag pairs. Functional tags are used to identify statement type (e.g. -Q(uestion)), to classify adjuncts (e.g. -

TMP temporal) and to indicate a basic distinction between subject and object grammatical function (-SBJ, -OBJ). CTB annotation implements phrasal projection and configurational marking of adjuncts and complements. For a detailed comparison between Penn-II (Marcus et al., 1994) and CTB annotation conventions see (Levy and Manning, 2003). An example CTB tree is given in Figure 1.

```
(IP-HLN
    (NP-PN-SBJ
        (NR 江泽民)
        (NR 李鹏))
    (VP
        (VV 电唁)
        (NP-OBJ
            (NP-PN
                (NR 尼克松))
            (NP
                (NN 逝世)))))
```

“江泽民李鹏电唁尼克松逝世”

“Jiang Zemin and Li Peng condoled the bereavement of Nixon by a telegram.”

Figure 1: Example CTB tree.

4 Automatic F-Structure Annotation Algorithm

4.1 Introduction

This section outlines the architecture of the automatic LFG f-structure annotation algorithm of (Cahill et al., 2002; Cahill et al., 2003; Cahill et al., 2004; O’Donovan et al., 2004). The generic algorithm is modular, as outlined in Figure 2, and is language- and treebank-independent. The modules of the annotation algorithm must be manually seeded with linguistic information for the specific treebank/language pair, in this case the CTB for Mandarin Chinese.

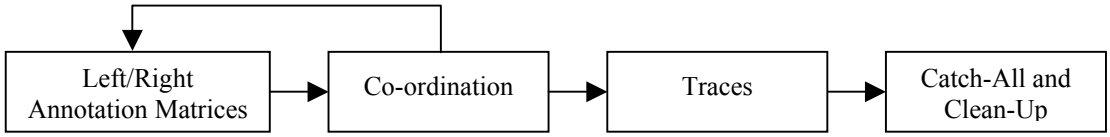

Figure 2: Annotation Algorithm Modules

The left-right context annotation matrices are based on a bi-partition of local trees of depth one (i.e. corresponding to CFG rules) into left context (Left) followed by the head (X) followed by right context (Right): XP → Left X Right. Each left-hand-side category XP is associated with an annotation matrix. For a given XP, the matrix states linguistic generalisations regarding the f-structure annotation of constituents to the left of the local head X and to the right of X. Annotation matrices are constructed by inspecting the most frequent rule types in a treebank expanding XP, so that the token occurrences of these rule types cover 85% of the corpus instances of XP expansions. In the case of the Penn-II treebank (Marcus et al., 1994) this means that instead of analysing >6000 NP rule types in the treebank, we only look at the most frequently occurring 102 NP rule types.

Co-ordination is treated in a separate component, as treebanks often encode co-ordination in a very flat manner. Separating out co-ordination in this way simplifies the statement of generalisations in the annotation matrices, supporting modularity and maintainability of the algorithm. The co-ordination component may reuse Left/Right annotation matrices to annotate local constituents outside the co-ordinate constituents of a parent category.

The Trace component of the algorithm exploits the treebank encoding on long-distance dependencies and translates such dependencies into corresponding re-entrancies in the f-structures. If the trace component is skipped, the resulting f-structures will be proto-f-structures, i.e. possibly partial representations of basic predicate-argument-adjunct structure with long-distance dependencies unresolved.

The first two components of the annotation algorithm sometimes overgeneralise to support the concise statement of linguistic generalisations. Such overgeneralisation is detected and corrected by the final Catch-All and Clean-Up component of the algorithm. Here we also provide a set of default annotations for any remaining unannotated nodes.

4.2 Seeding the LFG Annotation Algorithm with Chinese Linguistic Information

For a recent overview on LFG-based analyses of Chinese see e.g. (Bodomo and Luke, 2003). For LFG-based analyses of Cantonese Chinese see (Bodomo et al., 2004; Lam 2004). For LFG-based analyses of Mandarin Chinese see (Chief, 1996; Her, 2003; Sun, 2003).

The first module of the automatic f-structure annotation algorithm, Left-Right Annotation Matrices, head-lexicalises the CTB using the head-lexicalisation rules of (Levy and Manning, 2003). This process creates a bi-partition of each local subtree, with nodes lying in either the left or right context of the head. An annotation matrix is manually constructed for each parent category in the CTB. In order to seed the matrices, for each parent category in the CTB we extract the most frequent rule types expanding that category with joint coverage of >=85% of total rule token occurrences for the parent category. We distinguish between identical parent categories bearing different CTB functional tags. This results in 645 seed rule types in total. We then automatically provide partial annotations for these seed rule types based on the CTB functional tags found with daughter categories in the right-hand side of the rule types: to give a simple example, an –OBJ CTB tag triggers an ↑OBJ=↓ annotation. F-structure annotation of the partially automatically annotated seed rule types is then manually completed by the research team in Hong Kong. Annotation matrices are then constructed from the fully annotated seed rule set by the research team in Dublin.

The annotation of co-ordinate structures is handled by a separate module in the annotation algorithm, because the relatively flat analysis of co-ordination in the CTB would complicate the Left-Right Context Rules module, making them harder to maintain and extend. Once the elements of the co-ordination set have been identified, the Left-Right Context Rules module may be re-used to provide default annotations for any remaining unannotated nodes in a co-ordinate construction.

Currently our annotation algorithm does not include a trace component resolving long-distance dependencies, so that the annotation algorithm generates proto-f-structures for the CTB. Resolving long-distance dependencies in the manner of (Cahill et al., 2004) constitutes an avenue for future work.

The Catch-All and Clean-Up module provides default annotations for remaining unannotated nodes that are labelled with CTB functional tags. The functional tag –SBJ, for example, would be annotated ↑SUBJ=↓, while phrasal categories bearing –LOC or –TMP tags are annotated as elements of adjunct sets ↓∈↑ADJN. A small amount of over generation is accepted within the first two annotation algorithm modules to allow a concise statement of linguistic generalisations. Some annotations are overwritten to counter this problem and to systematically correct other potential feature clashes.

5 Annotation Algorithm Evaluation

The annotation algorithm is applied to each CTB tree and assigns functional annotations to nodes in CTB trees. The resulting annotations are collected, passed to a constraint solver and LFG f-structures are generated. The f-structures are evaluated for quantity and quality.

5.1 Quantitative Evaluation: Fragmentation

The annotation algorithm achieves good coverage for the CTB with 96.75% of the 3570 trees in the CTB training set (we follow the split into development, training and test sets in (Levy and Manning, 2003)) receiving a single connected and covering f-structure. Table 1 provides a quantitative evaluation of the f-structures produced by the annotation algorithm. Feature clashes in the annotation of 4 trees (0.112%) result in no f-structure being produced for those sentences. Nodes left unannotated by the annotation algorithm in 112 trees (3.137%) resulted in separate, disjoint f-structure fragments being produced for each of those sentences.

#Fragments	#Sentences	% Treebank
0	4	0.112
1	3454	96.751
2	105	2.941
3	4	0.112
4	1	0.028
7	1	0.028
9	1	0.028

Table 1: Annotation Coverage

5.2 Qualitative Evaluation against a Gold-Standard

While achieving such wide coverage is important, the annotation quality must be of a high standard, particularly as the annotation algorithm plays a vital role in the extraction of wide-coverage, probabilistic LFG parsing technology and lexical resources. Annotation quality is measured in terms of precision and recall against the dependencies derived from a set of manually constructed, gold-standard f-structures for 50 randomly selected sentences from the CTB test set. Following the methodology in (Cahill et al., 2002; King et al., 2003), the 50 CTB trees were automatically annotated with the f-structure annotation algorithm. The f-structure annotations were then manually corrected, extended and checked over a number of iterations.

Using the evaluation methodology and software presented in (Crouch et al., 2002) and (Riezler et al., 2002), the gold-standard f-structures and the f-structures generated by annotation algorithm were then translated into dependency triples and evaluated. Currently the automatic f-structure annotation algorithm achieves an f-score of 92.52% for complete f-structures and 85.92% for preds-only f-structures (Table 2).[1]

	All Grammatical Functions	Preds Only
Precision	92.41	85.96
Recall	92.63	85.88
F-Score	92.52	85.92

Table 2: Annotation Quality

[1] Preds-only f-structures consider only paths in f-structures ending in a **pred** feature-value pair.

Table 3 provides a breakdown of annotation results by feature name. Note that a number of features (classifier and obl) have been added manually to the gold-standard but are currently not supported by the automatic annotation algorithm, while obj2 is produced by the annotation algorithm but does not occur in the gold-standard.

	Precision	Recall	F-Score
adjunct	93	86	90
app	75	100	86
classifier	0	0	0
comp	23	39	23
coord	92	99	96
det	100	100	100
noun_type	100	100	100
number_type	33	67	44
obj	78	92	84
obj2	0	0	0
obl	0	0	0
pers	100	100	100
poss	98	90	94
quant	95	64	77
subj	91	87	89
topic	100	100	100
xcomp	80	80	80

Table 3: Annotation Quality Results by feature name

6 Extraction of Lexical Resources

In LFG, subcategorisation requirements are expressed at the level of f-structure and represented in terms of semantic forms. For example, a semantic form of type pred[subj,obj] states that the predicate pred locally requires a subj(ect) and an obj(ect) grammatical function. We refer to [subj,obj] as a frame type.

LFG distinguishes between subcategorisable (arguments: subj, obj, obj2, comp, xcomp etc.) and non-subcategorisable grammatical functions (adjuncts: adjn, xadjn, app etc.). If the f-structures generated by the automatic f-structure annotation algorithm on the treebank trees are of good quality, then reliable semantic forms can be extracted following the method presented in (O'Donovan et al., 2004): for each f-structure, for each level of embedding, determine the local pred and collect all subcategorisable grammatical functions present at that level (cf. Figure 3).

From the automatically f-structure-annotated CTB we extract a total of 10479 semantic form tokens with 26 distinct frame types. Of these 2510 are verbal semantic forms which occur with all 26 distinct frame types.

```
subj : coord_form : null
       coord : 1 : pred : '江泽民'
                   pers : 3
                   noun_type : proper
                   gloss : 'Jiang_Zemin'
```

```
                2 : pred : '李鹏'
                    pers : 3
                    noun_type : proper
                    gloss : 'Li_Peng'
pred : '电唁'
gloss : condole_by_a_telegram
obj : adjunct : 3 : pred : '尼克松'
                    pers : 3
                    noun_type : proper
                    gloss : 'Nixon'
      pred : '逝世'
      pers : 3
      noun_type : common
      gloss :  'bereavement'

Semantic form: 电唁([subj,obj])
```

"江泽民李鹏电唁尼克松逝世"
"Jiang Zemin and Li Peng condoled the bereavement of Nixon by a telegram."

Figure 3: An example of an automatically-generated f-structure and extracted semantic form.

	Tokens	Types
All forms	10469	26
Verbal	2510	26
Nominal	6227	4
Adjectival	715	1
Adverbial	579	1

Table 4: Semantic forms extracted from CTB

7 PCFG-Based LFG Approximations and Parsing Architectures

7.1 Methodology

Following the methodology presented in (Cahill et al., 2004) we extracted a number of PCFG-based LFG approximations in both the pipeline and integrated processing architectures.

In the pipeline architecture we first extract a PCFG from the CTB, use the PCFG to parse unseen text and send the trees generated for the unseen text to the automatic f-structure annotation algorithm to generate f-structures.

In the integrated architecture we first annotate the CTB with our automatic f-structure annotation algorithm, associating nodes in the treebank trees with one or more f-structure equations. We then extract an annotated PCFG (PCFG-A) where CFG categories (such as XP, YP, ZP) followed by (one or more) f-structure equations of the form [up… = down…] are interpreted as monadic categories for grammar extraction and parsing: XP[up… = down…] => YP[up… = down…] ZP[up… = down…]. We parse unseen text with the PCFG-A, retrieve the functional annotations from the parse trees and send them to a constraint solver to generate an f-structure.

The integrated architecture can, in fact, be understood as an instance of a grammar transformation (Johnson, 1999). In the case of a PCFG-A, the transformation is provided by the f-structure annotation algorithm.

In the experiments reported below, we furthermore study the effect of the parent transformation (Johnson, 1999) and its interaction with our two parsing architectures. The parent transformation annotates each non-preterminal node in a treebank tree with its parent category (thus encoding a limited, but useful, amount of contextual information not available to the original PCFG). In addition, we also study the effects of either preserving or deleting CTB functional tags in our extracted probabilistic grammars. CTB functional tags are different from the functional annotations (f-structure equations) generated by the f-structure annotation algorithm and consist of (possibly sequences of) functional tags of the form –TAG associated with CTB CFG categories.

In total, we extract the following probabilistic grammars:

- PCFG: a PCFG with all CTB functional tag annotations (F) stripped.
- PCFG-F: a PCFG with CTB functional tag annotations (F) preserved.
- PCFG-P: a PCFG with the parent transformation (P) but without CTB functional tags (F).
- PCFG-P-F: a PCFG with the parent transformation (P) and with CTB functional tags (F).
- PCFG-A: a PCFG without CTB functional tags (F) but with f-structure annotations (A).
- PCFG-A-P: a PCFG without CTB functional tags (F) but with f-structure annotations (A) and with parent transformation (P).

In each case, the experiments replicate the experimental set-up reported in (Levy and Manning, 2003) as regards split between training, development and test set. We use the BitPar parsing software (Schmid, 2004). Results of the parsing experiments are described and interpreted below.

7.2 Parsing Experiments

In order to assess the quality of the extracted grammars we carried out three types of parsing experiments:

- In experiment 1 we evaluate the CFG tree output of our parsers against the original trees for strings length <= 40 in articles 301-325 CTB, reporting f-scores for labelled and unlabelled bracketings using evalb.

- In experiment 2 we evaluate the f-structures generated by our grammars against the manually annotated 50 gold-standard f-structures for randomly selected trees from articles 301-325 using the triple-based dependency encoding and evaluation software from (Crouch et al., 2002; Riezler et al., 2002).

- In experiment 3 we evaluate the f-structures generated by our grammars against the f-structures for the full 318 test strings as generated by the automatic f-structure annotation algorithm for the *original* trees in articles 301-325 CTB using the triple-based dependency encoding and evaluation software from (Crouch et al., 2002; Riezler et al., 2002).

7.2.1 Experiment 1 (Tree-Based Evaluation)

Table 5 describes the results obtained in experiment 1. In this experiment we evaluate the parse output generated by our grammars against the original CTB trees in articles 301-325 (length <= 40) using

evalb, (cf. Sekine and Collins, 1997). Note that while coverage results in Table 5 are given for all 318 sentences (with no length restriction) in articles 301-325, f-scores are for the 271 sentences of length <= 40. We carry out the usual preprocessing steps prior to grammar extraction: deletion of empty nodes and cyclic unary productions (cf. Levy and Manning, 2003). PCFG is the grammar obtained by also deleting any CTB functional tags. PCFG-P is the parent-transformed PCFG (Johnson, 1999), while PCFG-A is the f-structure-annotated PCFG. Note the effect of the parent (P) and f-structure annotation (A) grammar transformations on grammar size. PCFG-F is the grammar extracted with CTB functional tags. PCFG-P (i.e. PCFG with parent transformation but without CTB functional tags) outperforms PCFG-F (i.e. the PCFG with CTB functional tags preserved) even though the size of PCFG-P is smaller than that of PCFG-F. This suggests that for Chinese and the given CTB tree representations, the parent transformation captures more pertinent information than the CTB functional tags. PCFG-P-F (i.e. PCFG with parent transformation and CTB functional tags) outperforms both PCFG-F and PCFG-P. Significantly, PCFG-A (f-structure annotations on the raw PCFG without CTB functional tags) outperforms PCFG-P-F (and, hence also PCFG-F and PCFG-P). Our best results achieved to date are those of the combined f-structure-annotated and parent transformed grammar PCFG-A-P with a labelled f-score of 81.57%, compared to the previous best reported labelled f-scores of 76.1% by (Hearne and Way, 2004), 78.8% by (Levy and Manning, 2003) and 79.9% by (Chiang and Bikel, 2002).

	PCFG	PCFG-F	PCFG-P	PCFG-P-F	PCFG-A	PCFG-A-P
#Rules	1498	3313	2611	6105	3224	6803
#Parses	318	318	318	318	318	317
Labelled F-Score	72.52	75.95	77.52	79.17	79.60	**81.57**
Unlabelled F-Score	73.25	77.08	78.20	80.00	80.23	**82.21**

Table 5: Parsing results for sentences of length <= 40 against articles 301-325

7.2.2. Experiment 2 (Dependency Evaluation against Gold-Standard)

Table 6 describes the results obtained in experiment 2. In this experiment we evaluate the f-structures generated by our grammars against the 50 gold-standard f-structures in terms of the triple encoding of dependencies and the evaluation software in (Crouch et al., 2002; Riezler et al., 2002). Compared to all grammatical functions, preds-only is the stricter measure as "minor" feature-value pairs such as those for (say) person features tend to be associated with the correct local pred even if the pred itself is misattached in the global f-structure (and corresponding dependency triple representation). It is interesting to note that even though there is a general tendency for grammars with better f-scores on trees (compare Table 5 above) to produce improved f-scores on dependency triples, the relative f-score dependency ranking between grammars deviates from that established on trees with PCFG-P-F and PCFG-A providing the best results against the gold-standard. The reason why PCFG-F and PCFG-P-F perform well in the pipeline architecture is that these grammars preserve CTB functional tags. These tags are exploited by the automatic annotation algorithm.

	PCFG	PCFG-F	PCFG-P	PCFG-P-F	PCFG-A	PCFG-A-P
All Grammatical Functions	66.56	79.55	66.77	82.79	81.22	76.99
Preds-only	52.30	61.36	52.17	67.74	64.80	62.35

Table 6: Parsing results against the gold-standard

7.2.3. Experiment 3 (Dependency Evaluation against Automatically Annotated Treebank Trees)

Table 7 describes the results obtained in experiment 3. In this experiment we evaluate the f-structures generated by our grammars against the f-structures generated by the automatic f-structure annotation algorithm for the *original* 318 treebank trees in the test set. Evaluation uses the triple encoding of dependencies and evaluation software of (Crouch et al., 2002; Riezler et al., 2002). Comparing Table 7

with Table 6 for experiment 2, above it is interesting to note that the dependency-based relative ranking in experiment 3 almost preserves the ranking established in experiment 2. The main difference is that that PCFG-A is now the best-performing grammar. Compared to experiment 2, overall results in experiment 3 are higher. This is to be expected: evaluation against a manually corrected and extended gold-standard is more taxing than evaluation against the automatically f-structure-annotated original treebank trees.

	PCFG	PCFG-F	PCFG-P	PCFG-P-F	PCFG-A	PCFG-A-P
All Grammatical Functions	66.84	83.53	67.38	85.39	86.06	82.36
Preds-only	54.78	69.40	56.27	72.75	73.98	71.09

Table 7: Parsing results for the sentences in articles 301-325 in CCG style experiment

8 Conclusions and Ongoing Work

In this paper we have reported on a project on inducing wide-coverage Lexical-Functional Grammar resources for Mandarin Chinese from treebanks. We estimate that to date we have spent less than a total of 3 person months between the research groups at Hong Kong and Dublin on the development of the automatic f-structure annotation algorithm for the CTB, the automatic extraction of wide-coverage PCFG-based LFG approximations, the extraction of lexical resources, the construction of a gold-standard for Chinese LFG resources and the evaluation experiments. In particular, the (partly) manual construction of a gold-standard for evaluation is non-trivial and time-consuming. We expect that our results to-date, while encouraging, can be improved significantly given further concerted research effort. In particular, we will continue working on refining the annotation algorithm, extending the gold-standard and including a treatment of long-distance dependencies along the lines presented for English in (Cahill et al., 2004) to generate proper rather than proto-f-structures for the CTB. Compared to our work on English (Cahill et al., 2004; O'Donovan et al., 2004) and German (Cahill et al. 2003), our work on Mandarin Chinese and the CTB to date uses a smaller feature set and a less fine-grained analysis. Currently 96.75% of the CTB trees receive a covering and connected f-structure, while 2.94% are associated with two f-structure fragments. From the f-structure-annotated CTB we extract a total of 12975 lexical entries with 20 distinct subcategorisation frame types. Of these, 3436 are verbal entries with a total of 11 different frame types. We extract a number of PCFG-based LFG approximations. Currently our best automatically-induced grammars achieve an f-score of 81.57% against the trees in unseen articles 301-325; 86.06% f-score (all grammatical functions) and 73.98% (preds-only) against the dependencies derived from the f-structures automatically generated from the original trees in 301-325 and 82.79% (all grammatical functions) and 67.74% (preds-only) against the dependencies derived from the manually-annotated gold-standard f-structures for 50 trees randomly selected from articles 301-325.The experiments and results reported here were carried out on a 4.1K sentence corpus, the CTB version 3.0 as described in (Xue et al, 2002). We will take this work as a seed to automatically annotate and induce LFG resources from the recently released full CTB with approximately 50K sentences. The results reported here and our previous experience with inducing wide-coverage LFG resources for English and German suggests that the treebank-based, constraint-based grammar induction method is attractive as it succeeds in generating multi-lingual wide-coverage resources at a much faster rate than traditional hand-coding of similar resources.

Acknowledgements

Our thanks go to Dr. Wu Hai at Dublin City University who helped with transliterating some of the Chinese gold-standard sentences into English for the Dublin group and to Prof. Chris Manning who provided us with the head-lexicalisation rules used in (Levy and Manning, 2003). The research reported here was partly supported by Enterprise Ireland Basic Research grant 04/BRG/CS0366, by an IBM Ph.D. scholarship 2004/5 for the first author, an IRCSET Ph.D. scholarship for the fifth author, an IBM visiting researcher fellowship for the seventh and an IBM visiting faculty fellowship for the

eighth author. Work on this project by the Hong Kong team was supported by the Simon K. Y. Lee Research Fund in Linguistics, Speech and Hearing Sciences on the project titled *International Workshop on Human Language Technology* (Project No.:21375232) and the Research Grants Council Direct Allocation Grant on the project titled *Complex Predicates and Serial Verbs Across Languages: Issues of syntax, semantics and information structure* (Project No.: 10204202).

References

Bodomo, A. and K. K. Luke. 2003. Lexical-Functional Grammar Analysis of Chinese. Editors: *Journal of Chinese Linguistics Monograph Series No. 19.*

Bodomo, A., O. S.-C. Lam, and N. S.-S. Yu. 2004. Double object and serial verb benefactive constructions in Cantonese. *Acta Orientalia*. [forthcoming]

Brants, T., S. Dipper, S. Hansen, W. Lezius and G. Smith. 2002. The TIGER Treebank. In E. Hinrichs and K. Simov, editors, *Proceedings of the first Workshop on Treebanks and Linguistic Theories (TLT'02)*, pages 24-41, Sozopol, Bulgaria.

Bresnan, J. 2001. *Lexical Functional Syntax*. Blackwell, Oxford.

Cahill, A., M. McCarthy, J. van Genabith and A. Way. 2002. Parsing Text with a PCFG Derived from Penn-II with an Automatic F-Structure Annotation Procedure. In M. Butt and T. Holloway-King, editors, *Proceedings of the 7th International Conference on Lexical-Functional Grammar*, pages 76-95, Athens, Greece. CSLI Publications, Stanford, CA.

Cahill, A., M. Forst, M. McCarthy, R. O'Donovan, C. Rohrer, J. van Genabith and A. Way. 2003. Treebank-Based Multilingual Unification Grammar Development. In *Proceedings of the 15th Workshop on Ideas and Strategies for Multilingual Grammar Development, at the 15th European Summer School in Logic, Language and Information,* pages 17-24, Vienna, Austria.

Cahill, A., M. Burke, R. O'Donovan, J. van Genabith and A. Way. 2004. Long-Distance Dependency Resolution in Automatically Acquired Wide-Coverage PCFG-Based LFG Approximations. In *Proceedings of the 42nd Conference of the Association for Computational Linguistics*, pages 320-327, Barcelona, Spain.

Charniak, E. 1996. Tree-bank Grammars. In *Proceedings of the 13th National Conference on Artificial Intelligence,* pages 1031-1036, Portland, OR. AAAI Press, Menlo Park, CA.

Chiang, D. and D. Bikel. 2002. Recovering Latent Information in Treebanks. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, pages 183-198, Taipei, Taiwan.

Chief, L-C. 1996 An LFG Account for Mandarin Reflexive Verbs. In Miriam Butt and Tracy Holloway King, editors, In *LFG Workshop: Proceedings of the First LFG Conference*, Rank Xerox Research Centre, Grenoble, France.

Crouch, R., R. Kaplan, T. King and S. Riezler. 2002. A Comparison of Evaluation Metrics for a Broad Coverage Parser. In *Beyond PARSEVAL Workshop at the 3rd International Conference on Language Resources and Evaluation (LREC 02)*, Las Palmas, Spain.

Hearne, M. and A. Way. 2004. Data-Oriented Parsing and the Penn Chinese Treebank. In *Proceedings of the First International Joint Conference on Natural Language Processing (IJCNLP-04)*, pages 406-413, Hainan Island, China.

Her, O-S. 2003. Chinese Inversion Constructions within a Simplified LMT. In *Journal of Chinese Linguistics Monograph Series No. 19*, pages 1-31.

Hockenmaier, J. 2003. Parsing with Generative Models of Predicate-Argument Structure. In *Proceedings of the 41st Annual Conference of the Association for Computational Linguistics*, pages 359-366, Sapporo, Japan.

Hockenmaier, J. and M. Steedman. 2002. Generative Models for Statistically Parsing with Combinatory Categorial Grammar. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 335-342, Philadelphia, PA.

Kaplan, R. and J. Bresnan, 1982. *The Mental Representation of Grammatical Relations*, chapter, Lexical-Functional Grammar: A Formal System for Grammatical Representations, pages 173-281. MIT Press, Cambridge, MA.

King, T. H., R. Crouch, S. Riezler, M. Dalrymple, R. M. Kaplan. 2003. The PARC 700 Dependency Bank, In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora at the 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1-8, Budapest, Hungary.

Lam, O. S.-C. 2004. *Order and Rank: Aspects of the Cantonese Verb Phrase*. M.Phil. Thesis. Department of Linguistics, the University of Hong Kong, Pokfulam. [forthcoming]

Levy, R. and C. Manning. 2003. Is it harder to parse Chinese, or the Chinese Treebank? In *Proceedings of the 41st Conference of the Association for Computational Linguistics,* pages 439-446, Sapporo, Japan.

Marcus, M., G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, M. Katz and B. Schasberger. 1994. The Penn Treebank: Annotating Predicate-Argument Structure. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 110-115, Princeton, NJ.

Miyao, Y., T. Ninomiya and J. Tsujii. 2003. Probabilistic Modeling of Argument Structures Including Non-Local Dependencies. In *Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP 2003),* Borovets, Bulgaria.

Miyao, Y., T. Ninomiya and J. Tsujii. 2004. Corpus-Oriented Grammar Development for Acquiring a Head-Driven Phrase Structure Grammar from the Penn Treebank. In *Proceedings of the First International Joint Conference on Natural Language Processing (IJCNLP-04)*, pages 390-397, Hainan Island, China.

O'Donovan, R., M. Burke, Cahill, A., J. van Genabith and A. Way. 2004. Large-Scale Induction and Evaluation of Lexical Resources from the Penn-II Treebank. In *Proceedings of the 42nd Conference of the Association for Computational Linguistics*, pages 367-374, Barcelona, Spain.

Pollard, C. and I. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago, IL.

Riezler, S., T.H. King, R.M. Kaplan, R. Crouch, J.T. Maxwell and M. 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 of Computational Linguistics (ACL 02)*. Philedelphia, PA. `

Schmid, H. 2004. Efficient Parsing of Highly-Ambiguous Context-Free Grammars with Bit Vectors. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, pages 162-168, Geneva, Switzerland.

Sekine, S and M. J. Collins. 1997. *The evalb Software*. http://nlp.cs.nyu.edu/evalb/

Sun, M. 2003. LFG for Chinese: Issues of Representation and Computation. In *Journal of Chinese Linguistics Monograph Series No. 19*, pages 129-151.

Xue, N., F-D. Chiou and M. Palmer. 2002. Building a Large-Scale Annotated Chinese Corpus. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan.