Accurate and Robust LFG-Based Generation 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

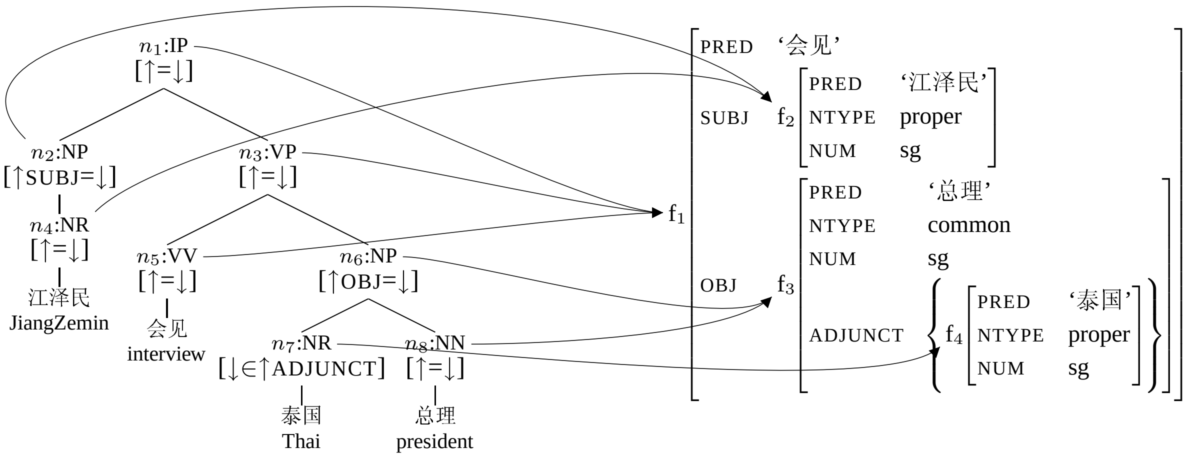
We describe three PCFG-based models for Chinese sentence realisation from Lexical-Functional Grammar (LFG) f-structures. Both the lexicalised model and the history-based model improve on the accuracy of a simple wide-coverage PCFG model by adding lexical and contextual information to weaken inappropriate independence assumptions implicit in the PCFG models. In addition, we provide techniques for lexical smoothing and rule smoothing to increase the generation coverage. Trained on 15,663 automatically LFG f-structure annotated sentences of the Penn Chinese treebank and tested on 500 sentences randomly selected from the treebank test set, the lexicalised model achieves a BLEU score of 0.7265 at 100% coverage, while the history-based model achieves a BLEU score of 0.7245 also at 100% coverage.

1 Introduction

Sentence generation, or surface realisation can be described as the problem of producing syntactically, morphologically, and orthographically correct sentences from a given abstract semantic / logical representation according to some linguistic theory, e.g. Lexical Functional Grammar (LFG), Head-Driven Phrase Structure Grammar (HPSG), Combinatory Categorial Grammar (CCG), Tree Adjoining Grammar (TAG) etc. Grammars, such as these, are declarative formulations of the correspondences between semantic and syntactic representations. Traditionally, grammar rules have been carefully handcrafted, such as those used in LinGo (Carroll et al., 1999), OpenCCG (White, 2004) and XLE (Crouch et al., 2007). As handcrafting grammar rules is time-consuming, language-dependent and domain-specific, recent years have witnessed research on extracting wide-coverage grammars automatically from annotated corpora, for both parsing and generation. FERGUS (Bangalore and Rambow, 2000) took dependency structures as inputs, and produced XTAG derivations by a stochastic tree model automatically acquired from an annotated corpus. Nakanishi et al. (2005) presented log-linear models for a chart generator using a HPSG grammar acquired from the Penn-II Treebank. From the same treebank, Cahill and van Genabith (2006) automatically extracted wide-coverage LFG approximations for a PCFG-based generation model.

In addition to applying statistical techniques to automatically acquire generation grammars, over the last decade, there has been a lot of interest in a generate-and-select paradigm for surface realisation. The paradigm is characterised by a separation between generation and selection, in which symbolic or rule-based methods are used to generate a space of possible paraphrases, and statistical methods are used to select one or more outputs from the space. Starting from Langkilde (2002) who used a n-gram language model to rank generated output strings, a substantial number of traditional handcrafted surface realisers have been augmented with sophisticated stochastic rankers (Velldal and Oepen, 2005; White et al., 2007; Cahill et al., 2007).

It is interesting to note that, while the study of how the granularity of context-free grammars (CFG) affects the performance of a parser (e.g. in the form



$$\phi : N \rightarrow F$$

$$\phi(n_1)=\phi(n_3)=\phi(n_5)=f_1 \quad \phi(n_2)=\phi(n_4)=f_2 \quad \phi(n_6)=\phi(n_8)=f_3 \quad \phi(n_7)=f_4$$

Figure 1: C- and f-structures with ϕ links for the sentence "江泽民会见泰国总理"

of grammar transforms (Johnson, 1998) and lexicalisation (Collins, 1997)) has attracted substantial attention, to our knowledge, there has been a lot less research on this subject for surface realisation, a process that is generally regarded as the reverse process of parsing. Moreover, while most of the research so far has concentrated on English or European languages, we are also interested in generation for other languages with diverse properties, such as Chinese which is currently a focus language in parsing (Bikel, 2004; Cao et al., 2007).

In this paper, we investigate three generative PCFG models for Chinese generation based on wide-coverage LFG grammars automatically extracted from the Penn Chinese Treebank (CTB). Our work is couched in the framework of Lexical Functional Grammar and is implemented in a chart-style generator. We briefly describe LFG and the basic generation model in Section 2. We improve the baseline PCFG model by weakening the independence assumptions in two disambiguation models in Section 3. Section 4 describes the smoothing algorithms adopted for the chart generator and Section 5 gives the experimental details and results.

2 LFG-Based Generation

2.1 Lexical Functional Grammar

Lexical Functional Grammar (Kaplan and Bresnan, 1982) is a constraint-based grammar formalism which postulates (minimally) two levels of representation: c(onstituent)-structure and f(unctional)-structure. C-structure takes the form of phrase structure trees and captures surface grammatical configurations. F-structure encodes more abstract grammatical functions (GFs) such as SUBJ(ect), OBJ(ect), ADJUNCT and TOPIC etc., in the form of hierarchical attribute-value matrices. C-structures and f-structures are related by a piecewise correspondence function ϕ that goes from the nodes of a c-structure tree into units of f-structure spaces (Kaplan, 1995). As illustrated in Figure 1, given a c-structure node n_i, the corresponding f-structure component f_j is $\phi(n_i)$. Admissible c-structures are specified by a context-free grammar. The corresponding f-structures are derived from functional annotations attached to the CFG rewriting rules.

(1) a. $\text{IP} \longrightarrow \underset{[\uparrow\text{SUBJ}=\downarrow]}{\text{NP}} \quad \underset{[\uparrow=\downarrow]}{\text{VP}}$

b. $\text{VP} \longrightarrow \underset{[\uparrow=\downarrow]}{\text{VV}} \quad \underset{[\uparrow\text{OBJ}=\downarrow]}{\text{NP}}$

c. $\text{NP} \longrightarrow \underset{[\uparrow\text{ADJ}=\downarrow]}{\text{NR}} \quad \underset{[\uparrow=\downarrow]}{\text{NN}}$

d. $\text{NP} \longrightarrow \underset{[\uparrow=\downarrow]}{\text{NR}}$

(1) shows a miniature set of annotated CFG rules (lexical entries omitted) which generates the c- and f-structure in Figure 1. In the functional annotations, (↓) refers to the f-structure associated with the local c-structure node n_i, i.e. $\phi(n_i)$, and (↑) refers to the

Model	Grammar Rule	Conditions
PCFG	VP[↑=↓] → VV[↑=↓] NP[↑OBJ=↓]	VP[↑=↓], {PRED, SUBJ, OBJ}
HB-PCFG	VP[↑=↓] → VV[↑=↓] NP[↑OBJ=↓]	VP[↑=↓], {PRED, SUBJ, OBJ}, TOP
LEX-PCFG	VP(会见)[↑=↓] → VV(会见)[↑=↓] NP(总理)[↑OBJ=↓]	VP(会见)[↑=↓], {PRED, SUBJ, OBJ}

Table 1: Examples of f-structure annotated CFG rules (from Figure 1) in different models

f-structure associated with the mother (M) node of n_i, i.e. $\phi(M(n_i))$.

2.2 Generation from f-Structures

The generation task in LFG is to determine which sentences correspond to a specified f-structure, given a particular grammar, such as (1). Kaplan and Wedekind (2000) proved that the set of strings generated by an LFG grammar from fully specified f-structures is a context-free language. Based on this theoretical cornerstone, Cahill and van Genabith (2006) presented a PCFG-based chart generator using wide-coverage LFG approximations automatically extracted from the Penn-II treebank. The LFG-based statistical generation model defines the conditional probability $P(T|F)$, for each candidate functionally annotated c-structure tree T (which fully specifies a surface realisation) given an f-structure F. The generation model searches for the T_{best} that maximises $P(T|F)$ (Eq. 1). $P(T|F)$ is then decomposed as the product of the probabilities of all the functionally annotated CFG rewriting rules $X \rightarrow Y$ (conditioned on the left hand side (LHS) X and local features of the corresponding f-structure $\phi(X)$) contributing to the tree T (Eq. 2). The first line (PCFG) of Table 1 shows the f-structure annotated CFG rule to expand node n_3 in Figure 1.

$$T_{best} = \underset{T}{\operatorname{argmax}}\, P(T|F) \qquad (1)$$

$$P(T|F) = \prod_{\substack{X \rightarrow Y in\, T \\ Feats = \{a_i | a_i \in \phi(X)\}}} P(X \rightarrow Y | X, Feats) \qquad (2)$$

3 Disambiguation Models

The basic generation model presented in (Cahill and van Genabith, 2006) used simple probabilistic context-free grammars. However, the independence assumptions implicit in PCFG models may not be appropriate to best capture natural language phenomena. Methodologies such as lexicalisation (Collins, 1997; Charniak, 2000) and tree transformations (Johnson, 1998), weaken the independence assumptions and have been applied successfully to parsing and shown significant improvements over simple PCFGs. In this section we study the effect of such methods in LFG-based generation for Chinese.

3.1 A History-Based Model

The history-based (HB) approach which incorporates more context information has worked well in parsing (Collins, 1997; Charniak, 2000). Resembling history-based models for parsing, Hogan et al. (2007) presented a history-based generation model to overcome some of the inappropriate independence assumptions in the basic generation model of (Cahill and van Genabith, 2006). The history-based model increases the context by simply including the parent grammatical function GF of the f-structure in addition to the local ϕ-linked feature set in the conditioning context (Eq. 3). The f-structure annotated CFG rule expanding n_3 in the history-based model is shown in the second line (HB-PCFG) of Table 1.[1]

$$P(T|F) = \prod_{\substack{X \rightarrow Y in\, T \\ Feats = \{a_i | a_i \in \phi(X)\} \\ \exists f\ (f\ \mathbf{GF}) = \phi(X)}} P(X \rightarrow Y | X, Feats, \mathbf{GF}) \qquad (3)$$

The history-based model is motivated by English data, for example, to generate the appropriate case for pronouns in subject position and object position, respectively. Though Chinese does not distinguish cases, we expect the f-structure parent GF to help predict grammar rule expansions more accurately in the tree derivation than the simple PCFG model. We will investigate how the HB model performs while migrating it from English to Chinese data.

[1] The parent grammatical function of the outermost f-structure is assumed to be a dummy GF TOP.

3.2 A Lexicalised Model

Compared to the HB model which includes the parent grammatical function in the conditioning context, lexicalised grammar rules contain more fine-grained categorial information. To the best of our knowledge, lexicalised parsers (Bikel, 2004) outperform unlexicalised parsers for Chinese. The expectation is that a lexicalised PCFG model also works better than a simple PCFG model in Chinese generation, considering e.g. prepositional phrase (PP) modification in Chinese. Some prepositions indicating directions can occur either before or after the main verbs, for instance both (2a) and (2b) are acceptable in Chinese. However, most PP modifiers only act as adverbial adjuncts between the subjects and verbal predicates. For instance "对/to" never follows a verb as exemplified in the ungrammatical sentence (3b).

(2) a. 这 趟 列车 开 **往北京**
this CLS train run **to Beijing**
'The train is bound for Beijing.'

b. 这 趟 列车 **往北京** 开
this CLS train **to Beijing** run

(3) a. 泰国 总理 **对中国** 进行 访问
Thai president to China make visit
'The Thai president paid a visit to China.'

b. * 泰国 总理 进行 访问 **对中国**
Thai president make visit to China

In order to model phenomena such as these, we head-lexicalise our grammar by associating each non-terminal node with the head word[2] in the c-structure tree along the head-projection line. A non-terminal node is written as $X(x)$, where x is the lexical head of X. The example generation grammar rule in the lexicalised model is shown in the last line (LEX-PCFG) of Table 1.

As in CKY chart parsing, generation grammars are binarised in our chart generator. Thus all grammar rules are either unary of the form $X \rightarrow H$ or binary $X \rightarrow YH$ (or $X \rightarrow HY$), where H is the head constituent and Y is the modifier. To handle the problem of sparse data while estimating rule probabilities, a back-off to baseline model is employed. As, from a linguistic perspective, it is the modifier rather than the head word which plays the main role in determining word order, a back-off to partial lexicalisation on the modifier only is also used for binary rules. As a result, the probabilities of lexicalised unary and binary CFG rules are calculated as in Eq. (4) and Eq. (5), respectively.

$$\begin{aligned} P_{bk}(H(h)|X(h)) &= \lambda_1 P(H(h)|X(h)) \\ &\quad + \lambda_2 P(H|X) \end{aligned} \tag{4}$$

$$\begin{aligned} P_{bk}(Y(y)H(h)|X(h)) &= \lambda_1 P(Y(y)H(h)|X(h)) \\ &\quad + \lambda_2 P(Y(y)H|X) + \lambda_3 P(YH|X) \end{aligned} \tag{5}$$

$$where \quad \sum_{i=1} \lambda_i = 1$$

In principle, grammars binarisation from left-to-right (left-) or from right-to-left (right-) are equivalent to represent the original grammar and the probability distributions. However the head word is the final constituent for most phrasal categories in Chinese.[3] In lexicalised model, the head word immediately projects to the top level in a left-binary tree, and as a result, the intermediate $\overline{\mathrm{NP}}$ nodes cannot be lexicalised with the head word as illustrated in Figure (2b). By contrast, right-binary rules are lexicalised and the head word is percolated from the bottom of the tree (Figure (2c)). Therefore we adopt the right binarisation method in our generation algorithm.

4 Chart Generation and Smoothing Algorithms

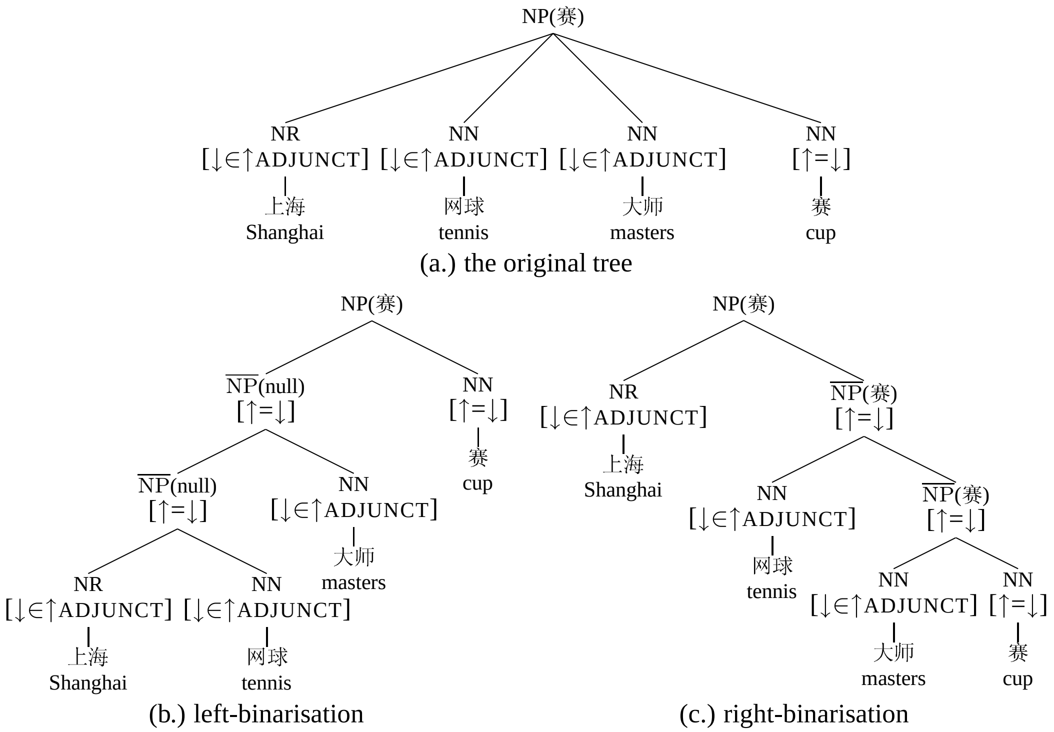
4.1 Chart Generation Algorithm

The PCFG-based generation algorithms are implemented in terms of a chart generator (Kay, 1996). In the generation algorithm, each (sub-)f-structure indexes a (sub-)chart. Each local chart generates the most probable trees for the local f-structure in a bottom-up manner:

- generating lexical edges from the the local GF PRED and some atomic features representing function words, mood or aspect etc.

[2] We use a mechanism similar to (Collins, 1997) but adapted to Chinese data to find lexical heads in the treebank data.

[3] Except for prepositional phrases, localiser and some verbal phrases.



(a.) the original tree

(b.) left-binarisation

(c.) right-binarisation

Figure 2: Lexicalised binary trees

- applying unary rules and binary rules to generate new edges until no any new edges can be generated in the current local chart.
- propagating compatible edges to the upper-level chart.

For efficiency, the generation algorithm does Viterbi-pruning for each local chart, viz. if two edges have equivalent categories and lexical coverage, only the most probable one is kept.

The generation coverage is impacted on by unknown words[4] and unmatched grammar rules in chart generation. We present a lexical smoothing and a rule smoothing strategy in the following subsections.

4.2 Lexical Smoothing

In LFG f-structure, the surface form of the lemma is represented via lexical rules involving a particular set of features, e.g. the lemma “总理/president” is represented as {↑PRED=‘总理’, ↑NTYPE=common, ↑NUM=sg}. Particular lexical rules can be captured in general lexical macros abstracting away from particular surface forms to lemmas, e.g. the lexical macro encapsuling the above lexical rule is {↑PRED=$LEMMA, ↑NTYPE=common, ↑NUM=sg}, which generally associates to common nouns NN in the CTB. According to the assumption that unknown words have a probability distribution similar to hapax legomenon (Baayen and Sproat, 1996), we predict the part-of-speech of unknown words from infrequent words in the training set by automatically extracting lexical macros corresponding to the particular set of f-structure features. The probability of the potential POS tag t associated to a feature set f is estimated according to Eq. (6).

$$P(t|f) = \frac{count(t,f)}{\sum_{i=1}^{n} count(t_i,f)} \tag{6}$$

4.3 Rule Smoothing

The coverage of grammar rules increases with the size of training data and in theory all the rules can be fully covered by a training set, if it is big enough. With limited training resources we have to resort to fuzzy matching of grammar rules. Two smoothing strategies are carried out at the level of grammar rules.

[4] We use unknown words as a cover term to refer to all words occurring in the test set but not in the training set.

	Mathched Grammar Rule
Nonsmooth	VP[↑=↓] → VV[↑=↓] NP[↑OBJ=↓], {SUBJ, OBJ, PRED}
Feature smooth	VP[↑=↓] → VV[↑=↓] NP[↑OBJ=↓]
Partial match	VP → VV [↑OBJ=↓], {SUBJ, OBJ, PRED}

Table 2: Smoothing of CFG rules

- Reducing the conditioning f-structure features during rule matching;
- Applying partial match during rule application.

A node in each unlexicalised grammar rule $X \rightarrow YH$ includes two parts: constituent category c, such as IP, NP, VP etc.; functional f-structure annotation a, such as [↑SUBJ=↓], [↑=↓] etc. As a heuristic based on linguistic experience, we define the order of importance of these elements as follows:

$$X(c) > H(c) > Y(a) > Y(c) > X(a) > H(a)$$

(4) IP[↑COMP=↓] → NP[↑SUBJ=↓] VP[↑=↓]

For the above example rule (4), the importance of the elements is:

IP > VP > [↑SUBJ=↓] > NP > [↑COMP=↓] > [↑=↓]

The elements can be deleted from the rules in an importance order from low to high.[5] The partial rules adopted in our system ignore the least important 3 elements, viz. the functional annotation of the head node $H(a)$, the functional annotation on LHS $X(a)$ and constituent category of the modifier node $Y(c)$. Examples of the two types of smoothed rules are shown in Table 2.

5 Experimental Results

Our experiments are carried out on the newly released Penn Chinese treebank version 6.0 (CTB6) (Xue et al., 2005), excluding the portion of ACE broadcast news. We follow the recommended splits (in the list-of-file of CTB6) to divide the data into test set, development set and training set. The training set includes 756 files with a total of 15,663 sentences. The CTB trees of the training set were automatically annotated with LFG f-structure equations following (Guo et al., 2007). Table 3 shows the number of different grammar rule types extracted from the training set. From the test files, we randomly select 500 sentences as test data with minimal sentence length 5 words, maximal length 80 words, and average length 28.84 words. The development set also includes 500 sentences randomly selected from the development files with sentence length between 5 and 80 words. The c-structure trees of the test and development data were also automatically converted to f-structures as input to the generator.

Type	with features	without features
PCFG	22,372	8,548
HB-PCFG	28,487	11,969
LEX-PCFG	325,094	286,468

Table 3: Number of rules in the training set

The generation system is evaluated against the raw text of the test data in terms of accuracy and coverage. Following (Langkilde, 2002) and other work on general-purpose generators, we adopt BLEU score (Papineni et al., 2002), average simple string accuracy (SSA) and percentage of exactly matched sentences for accuracy evaluation.[6] For coverage evaluation, we measure the percentage of input f-structures that generate a sentence.

Table 4 reports the initial experiments on the simple PCFG, HB-based PCFG and lexicalised PCFG models. The results in the left column evaluate all input f-structures, the right column evaluate only those f-structures which yield a complete sentence. The results show that the lexicalised model outperforms the baseline PCFG model. The HB model is the most accurate for complete sentences, but with reduced coverage compared to the other two models. However the low coverage of sentences completely generated due to unknown words and unmatched rules makes the results unusable in prac-

[5] However c and a on the same node can't be deleted at the same time.

[6] We are aware of the limitations in fully automatic evaluation metrics, and in an ideal scenario, we would complement the BLEU and SSA scores by a human evaluation. Unfortunately, this is beyond the scope of the current paper.

	All Output Strings				Complete Output Sentences			
	Coverage	ExMatch	BLEU	SSA	Coverage	ExMatch	BLEU	SSA
PCFG	100%	7.2%	0.5401	0.6261	36.40%	19.78%	0.7101	0.7687
HB-PCFG	100%	8.60%	0.5474	0.6281	34.80%	24.71%	**0.7513**	**0.8092**
LEX-PCFG	100%	**9.40%**	**0.5687**	**0.6537**	37.00%	**25.41%**	0.7431	0.8024

Table 4: Results without smoothing

	All Output Strings				Complete Output Sentences			
	Coverage	ExMatch	BLEU	SSA	Coverage	ExMatch	BLEU	SSA
PCFG	100%	11.00%	0.6894	0.7240	94.20%	11.68%	0.7047	0.7388
HB-PCFG	100%	11.80%	0.7108	0.7348	94.00%	12.55%	0.7284	0.7506
LEX-PCFG	100%	**14.00%**	**0.7152**	**0.7595**	94.40%	**14.83%**	**0.7302**	**0.7754**

Table 5: Results with lexical smoothing

	Partial match				Feature smooth			
Complete Sentences	Coverage	ExMatch	BLEU	SSA	Coverage	ExMatch	BLEU	SSA
PCFG	97.20%	11.32%	0.7022	0.7356	100%	11.20%	0.7021	0.7330
HB-PCFG	96.20%	12.27%	0.7263	0.7458	100%	12.00%	0.7245	0.7413
LEX-PCFG	97.80%	**14.31%**	**0.7265**	**0.7696**	100%	**14.20%**	**0.7265**	**0.7675**

Table 6: Results with lexical and rule smoothing

tice.

Table 5 gives the results with lexical smoothing. The coverage for complete sentences increases by nearly 60% absolute for all models. The increased coverage also improves the overall results evaluated against all sentences. The HB model performs better than the simple PCFG model in nearly all respects and in turn the lexicalised model comprehensively outperforms the HB model.

The final results with both lexical smoothing and rule smoothing by two different strategies are tabulated in Table 6. The left column provides the results of smoothing by partial match and the right column the results by reducing conditioning f-structure features. All results are evaluated for completely generated sentences only. The feature smoothing results in a full coverage of 100%, while slightly degrading the quality of sentences generated compared with partial match smoothing. We feel the tradeoff at the cost of a small decrease in quality is still worth the full coverage. Throughout the experiments, the lexicalised model exhibits consistently better performance than the unlexicalised models, which proves our intuition that successful techniques in parsing also work well in generation.

6 Conclusion and Further Work

We have presented an accurate, robust chart generator for Chinese based on treebank-based, automatically acquired LFG resources. Our model improves the baseline provided by (Cahill and van Genabith, 2006): (i) accuracy is increased by creating a lexicalised PCFG grammar and enriching conditioning context with parent f-structure features; and (ii) coverage is increased by providing lexical smoothing and fuzzy matching techniques for rule smoothing.

The combinational explosion of grammar rules encountered in the chart generator is similar to that in parsing. In the current system, we only keep the most probable realisation for each input f-structure. An alternative model in line with the generate-and-select paradigm, would pack all the locally equivalent edges in a forest and re-rank all the realisations by a separate language model. This might help us to reduce some errors caused in our current model, for instance, the generation of function words in fixed phrases. As shown in ex. (5), the function word “之” is incorrectly generated as “的”. This is because they share the same part-of-speech DEG in CTB, however “的” has a much higher frequency than “之” in Chinese text and thus has a higher probability to be generated.

(5) a. 万物 之 中
all things DE in
'among all things'

b. * 万物 **的** 中
all things DE in

Acknowledgments

The research reported in this paper is supported by Science Foundation Ireland grant 04/IN/I527. Also, we would like to thank Aoife Cahill for many helpful and insightful discussions on the work. And we gratefully acknowledge the anonymous reviewers.

References

Harald Baayen and Richard Sproat. 1996. Estimating Lexical Priors for Low-Frequency Morphologically Ambiguous Forms. *Computational Linguistics*, 22(2): 155–166.

Srinivas Bangalore and Owen Rambow. 2000. Exploiting a Probabilistic Hierarchical Model for Generation. *Proceedings of the 18th International Conference on Computational Linguistics*, pages 42–48. Saarbrücken, Germany.

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

Aoife Cahill and Josef van Genabith. 2006. Robust PCFG-Based Generation Using Automatically Acquired LFG Approximations. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1033–1040. Sydney, Australia.

Aoife Cahill and Martin Forst and Christian Rohrer. 2007. Stochastic Realisation Ranking for a Free Word Order Language. *Proceedings of the 11th European Workshop on Natural Language Generation*, pages 17–24. Schloss Dagstuhl, Germany.

John Carroll and Ann Copestake and Dan Flickinger and Victor Poznanski. 1999. An efficient chart generator for (semi-)lexicalist grammars. *Proceedings of the 7th European Workshop on Natural Language Generation*, pages 86–95. Toulouse, France.

Hailong Cao and Yujie Zhang and Hitoshi Isahara. 2007. Empirical study on Parsing Chinese Based on Collins' Model. *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pages 113–119. Melbourne, Australia.

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

Michael Collins. 1997. Three Generative, Lexicalized Models for Statistical Parsing. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 16–23. Madrid, Spain.

Dick Crouch and Mary Dalrymple and Ron Kaplan and Tracy King and John Maxwell and Paula Newman. 2007. XLE Documentation. Palo Alto Research Center, CA.

Yuqing Guo and Josef van Genabith and Haifeng Wang. 2007. Treebank-based Acquisition of LFG Resources for Chinese. *Proceedings of LFG07 Conference*, pages 214–232. Stanford, CA, USA.

Deirdre Hogan and Conor Cafferkey and Aoife Cahill and Josef van Genabith. 2007. Exploiting Multi-Word Units in History-Based Probabilistic Generation. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 267–276. Prague, Czech Republic.

Mark Johnson. 1998. PCFG Models of Linguistic Tree Representations. *Computational Linguistics*, 24(4): 613–632. MIT Press, Cambridge, MA,

Ronald M. Kaplan. 1995. The formal architecture of lexical-functional grammar. *Formal Issues in Lexical-Functional Grammar*, pages 7–27. CSLI Publications, Standford, USA.

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.

Ronald M. Kaplan and Jurgen Wedekind. 2000. LFG Generation Produces Context-free Languages. *Proceedings of the 18th International Conference on Computational Linguistics*, pages 425–431. Saarbrücken, Germany.

Martin Kay. 1996. Chart Generation. *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 200–204. Santa Cruz, USA.

Irene Langkilde. 2000. Forest-Based Statistical Sentence Generation. *Proceedings of 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 170–177. Seattle, WA.

Langkilde, Irene. 2002. An Empirical Verification of Coverage and Correctness for a General-Purpose Sentence Generator. *Proceedings of the Second International Conference on Natural Language Generation*, 17–24. New York, USA.

Hiroko Nakanishi and Yusuke Nakanishi and Jun'ichi Tsujii. 2005. Probabilistic Models for Disambiguation of an HPSG-Based Chart Generator. *Proceedings of the 9th International Workshop on Parsing Technology*, pages 93–102. Vancouver, British Columbia.

Kishore Papineni and Salim Roukos and Todd Ward and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311-318. Philadelphia, USA.

Erik Velldal and Stephan Oepen. 2005. Maximum entropy models for realization ranking. *Proceedings of the MTSummit '05*.

Michael White. 2004. Reining in CCG Chart Realization. *Proceedings of the third International Natural Language Generation Conference*. Hampshire, UK.

Michael White and Rajakrishnan Rajkumar and Scott Martin. 2007. Towards Broad Coverage Surface Realization with CCG. *Proceedings of the MT Summit XI Workshop on Language Generation and Machine Translation*, pages 22–30. Copenhagen, Danmark.

Nianwen Xue and Fei Xia and Fu dong Chiou and Martha Palmer. 2005. The Penn Chinese TreeBank: Phrase Structure Annotation of a Large Corpus. *Natural Language Engineering*, 11(2): 207–238.