The Challenges of Parsing Chinese with Combinatory Categorial Grammar

Daniel Tse and **James R. Curran**
School of Information Technologies
University of Sydney
Australia
{dtse6695,james}@it.usyd.edu.au

Abstract

We apply Combinatory Categorial Grammar to wide-coverage parsing in Chinese with the new Chinese CCGbank, bringing a formalism capable of transparently recovering non-local dependencies to a language in which they are particularly frequent.

We train two state-of-the-art English CCG parsers: the parser of Petrov and Klein (P&K), and the Clark and Curran (C&C) parser, uncovering a surprising performance gap between them not observed in English — 72.73 (P&K) and 67.09 (C&C) F-score on PCTB 6.

We explore the challenges of Chinese CCG parsing through three novel ideas: developing corpus variants rather than treating the corpus as fixed; controlling noun/verb and other POS ambiguities; and quantifying the impact of constructions like *pro*-drop.

1 Introduction

Automatic corpus conversions from the Penn Treebank (Marcus et al., 1994) have driven research in lexicalised grammar formalisms, such as LTAG (Xia, 1999), HPSG (Miyao et al., 2004) and CCG (Hockenmaier and Steedman, 2007), producing the lexical resources key to wide-coverage statistical parsing.

The Chinese Penn Treebank (PCTB; Xue et al., 2005) has filled a comparable niche, enabling the development of a Chinese LTAG (Xia et al., 2000), a wide-coverage HPSG parser (Yu et al., 2011), and recently Chinese CCGbank (Tse and Curran, 2010), a 750 000-word corpus of Combinatory Categorial Grammar (CCG; Steedman, 2000) derivations.

We train two CCG parsers, Clark and Curran (C&C; 2007), and the Petrov and Klein (P&K; 2007) PCFG parser, on Chinese CCGbank. We follow Fowler and Penn (2010), who treat the English CCGbank (Hockenmaier and Steedman, 2007) grammar as a CFG and train and evaluate the P&K parser directly on it.

We obtain the first Chinese CCG parsing results: F-scores of 72.73 (P&K) and 67.09 (C&C) on labelled dependencies computed over the PCTB 6 test set. While the state-of-the-art in Chinese syntactic parsing has always lagged behind English, this large gap is surprising, given that Fowler and Penn (2010) found only a small margin separated the two parsers on English CCGbank (86.0 versus 85.8).

Levy and Manning (2003) established that properties of Chinese such as noun/verb ambiguity contribute to the difficulty of Chinese parsing. We focus on two factors within our control: annotation decisions and parser architecture.

Existing research has varied parsers whilst keeping the corpus fixed. We vary the corpus whilst keeping the parsers fixed by exploring multiple design choices for particular constructions. By exploiting the fully automatic CCGbank extraction process, we can immediately implement these choices and assess their impact on parsing performance.

Secondly, we contrast the performance of C&C, with its tagging/parsing pipeline, with P&K, a parser which performs joint tagging and parsing, and establish that P&K is less sensitive to the greater lexical category ambiguity in Chinese CCGbank.

We demonstrate that Chinese CCG parsing is very difficult, and propose novel techniques for identifying where the challenges lie.

2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 295–304,
Montréal, Canada, June 3-8, 2012.

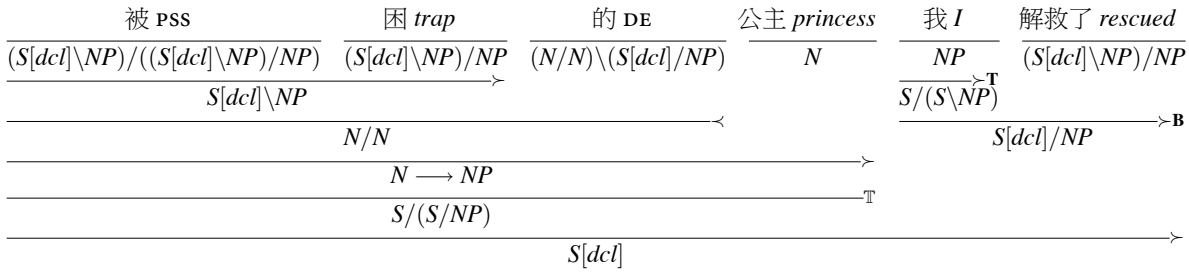

Figure 1: 3 types of non-local dependencies in 6 words: "*(As for) the trapped princess, I rescued (her)*."

2 Background

Bikel and Chiang (2000) developed the first PCTB parser, demonstrating that Chinese was similar enough to English for techniques such as a Collins-style head-driven parser or TAG to succeed. Later PCTB parsers used Tree Insertion Grammar (Chiang and Bikel, 2002), PCFGs (Levy and Manning, 2003), the Collins models (Bikel, 2004) and transition-based discriminative models (Wang et al., 2006; Zhang and Clark, 2009; Huang et al., 2009). These systems also established the relative difficulty of parsing Chinese and English; while PARSEVAL scores over 92% are possible for English (McClosky et al., 2006), systems for Chinese have achieved only 87% (Zhang and Clark, 2009) on the same metric.

Non-local dependencies (NLDs) are lexical dependencies which hold over unbounded distances. Guo et al. (2007) observed that despite the importance of NLDs for correct semantic interpretation, and the fact that Chinese syntax generates more NLDs than English, few parsers in Chinese are equipped to recover the traces which mark NLDs. For instance, extraction, a common NLD type, occurs more frequently in CPTB sentences (38%) compared to PTB (17%).

A more satisfying approach is to use a grammar formalism, such as CCG (Steedman, 2000), which generates them inherently, enabling a unified parsing model over local and non-local dependencies. This approach is taken in the C&C parser (Clark and Curran, 2007), which can directly and transparently recover NLDs in English (Rimell et al., 2009).

Chinese CCGbank (Tse and Curran, 2010) demonstrates that a parsimonious account of Chinese syntax with CCG is possible. Many familiar objects of Chinese syntax which generate NLDs, including the 把 *ba*/被 *bei* constructions, topicalisation and extraction receive natural CCG analyses in Chinese

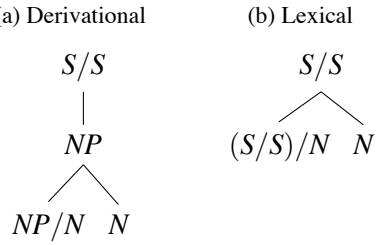

Figure 2: Two types of ambiguity

CCGbank. Figure 1 shows the CCGbank analysis of passivisation, topicalisation and extraction, creating NLDs between 公主 *princess* and each of 被 *PSS*, 困 *trap* and 解救 *rescue* respectively.

We take two state-of-the-art parsers and train them to establish the difficulty of parsing Chinese with CCG. The first is the Clark and Curran (C&C; 2007) parser, which uses *supertagging* (Clark and Curran, 2004), a local, linear-time tagging technique which drastically prunes the space of lexical categories which the polynomial-time parsing algorithm later considers. The second is the *coarse-to-fine* parser of Petrov and Klein (2007) which iteratively refines its grammar by splitting production rules to uncover latent distinctions. Fowler and Penn (2010) demonstrate that the English CCGbank grammar is strongly context-free, allowing them to treat it as a CFG and train the Petrov and Klein (2007) parser directly.

2.1 Derivational vs. lexical ambiguity

The designer of a CCGbank must frequently choose between derivational and lexical ambiguity (Hockenmaier, 2003; Tse and Curran, 2010). *Derivational ambiguity* analyses special constructions through arbitrary label-rewriting phrase structure rules, while *lexical ambiguity* assigns additional categories to lexical items for when they participate in special constructions.

Derivational and lexical ambiguity often arise in CCG because of the *form-function distinction* — when the syntactic *form* of a constituent does not coincide with its semantic *function* (Honnibal, 2010). For instance, in English, topicalisation causes an *NP* to appear in clause-initial position, fulfilling the *function* of a sentential pre-modifier while maintaining the *form* of an *NP*. Figure 2 shows two distinct CCG analyses which yield the same dependency edges.

Derivational ambiguity increases the parser search space, while lexical ambiguity enlarges the tag set, and hence the complexity of the supertagging task.

3 Three versions of Chinese CCGbank

We extract three versions of Chinese CCGbank to explore the trade-off between lexical and derivational ambiguity, training both parsers on each corpus to determine the impact of the annotation changes. Our hypothesis is that the scarcity of training data in Chinese means that derivational ambiguity results in better coverage and accuracy, at the cost of increasing time and space requirements of the resulting parser.

3.1 The lexical category *LC* (localiser)

In the following sentences, the words in bold have often been analysed as belonging to a lexical category *localiser* (Chao, 1968; Li and Thompson, 1989).

(1) a. 屋子 里面
house **inside**:LC
the inside of the house/inside the house
b. 大 树 旁边
big tree **beside**:LC
(the area) beside the big tree

Localisers, like English prepositions, identify a (temporal, spatial, etc.) extent of their complement. However, the combination *Noun + Localiser* is ambiguous between noun function (*the inside of the house*) and modifier function (*inside the house*).

We consider two possibilities to represent localisers in CCG, which trade derivational for lexical ambiguity. In (2-a), a direct CCG transfer of the PCTB analysis, the preposition 在 *at* expects arguments of type *LCP*. In (2-b), 在 *at* now expects only *NP* arguments, and the unary promotion $LCP \rightarrow NP$ allows *LCP*-form constituents to *function* as *NP*s.

(2) a.
$$\dfrac{\dfrac{\text{在 at}}{PP/LCP} \quad \dfrac{\dfrac{\text{房子 room}}{NP} \quad \dfrac{\text{里 in:LC}}{LCP\backslash NP}\prec}{LCP}}{PP}\succ$$

b.
$$\dfrac{\dfrac{\text{在 at}}{PP/NP} \quad \dfrac{\dfrac{\text{房子 room}}{NP} \quad \dfrac{\text{里 in:LC}}{LCP\backslash NP}\prec}{LCP \rightarrow NP}}{PP}\succ$$

The analysis in (2-a) exhibits greater lexical ambiguity, with the lexical item 在 *at* carrying at least two categories, PP/NP and PP/LCP, while (2-b) trades off derivational for lexical ambiguity: the unary promotion $LCP \rightarrow NP$ becomes necessary, but 在 *at* no longer needs the category PP/LCP.

The base release of Chinese CCGbank, corpus **A**, like (2-a), makes the distinction between categories *LCP* and *NP*. However, in corpus **B**, we test the impact of applying (2-b), in which the unary promotion $LCP \rightarrow NP$ is available.

3.2 The bare/non-bare *NP* distinction

The most frequent unary rule in English CCGbank, occurring in over 91% of sentences, is the promotion from bare to non-bare nouns: $N \rightarrow NP$. Hockenmaier (2003) explains that the rule accounts for the form-function distinction in determiner-less English nouns which nevertheless have definite reference, while preventing certain over-generations (e.g. *the the car). The *N-NP* distinction also separates adjectives and noun modifiers (category N/N), from predeterminers (category NP/NP) (Hockenmaier and Steedman, 2005), a distinction also made in Chinese.

While Chinese has strategies to mark definite or indefinite reference, they are not obligatory, and a bare noun is referentially ambiguous, calling into question whether the distinction is justified in CCG:

(3) a. 狗 很 聪明
dog very clever
Dogs are clever.
b. 我 看到 狗
1SG see dog
I saw a dog/dogs.
c. 狗 跑走 了
dog run-away ASP
The dog/dogs ran away.

The fact that the Chinese determiner is not necessarily a maximal projection of the noun – in other words, the determiner does not 'close off' a level of *NP* – also argues against importing the English analysis. In contrast, the English CCGbank determiner category *NP*/*N* reflects the fact that determiners 'close off' *NP* — further modification by noun modifiers is blocked after combining with a determiner.

(4) 共和党 这 举动
Republican Party this act
this action by the Republican Party

To test its impact on Chinese parsing, we create a version of Chinese CCGbank (corpus **C**) which neutralises the distinction. This eliminates the atomic category *N*, as well as the promotion rule $N \rightarrow NP$.

4 Experiments

While a standard split of PCTB 5 exists, as defined by Zhang and Clark (2008), we are not aware of a consistently used split for PCTB 6. We present a new split in Table 1 which adds data from the ACE broadcast section of PCTB 6, maintaining the same train/dev/test set proportions as the PCTB 5 split.

We train C&C using the *hybrid* model, the best-performing model for English, which extracts features from the dependency structure (Clark and Curran, 2007). We use $\beta = \langle 0.055, 0.01, 0.05, 0.1 \rangle$ during training with a Gaussian smoothing parameter $\alpha = 2.4$ (optimised on the corpus **A** dev set). We use $\beta = \langle 0.15, 0.075, 0.03, 0.01, 0.005, 0.001 \rangle$ during parsing, with the maximum number of supercats (chart entries) set to 5,000,000, reflecting the greater supertagging ambiguity of Chinese parsing.

The P&K parser is used "off-the-shelf" and trained with its default parameters, only varying the number of split-merge iterations and enabling the Chinese-specific lexicon features. The P&K parser involves no explicit POS tagging step, as the (super)tags correspond directly to non-terminals in a CFG.

Fowler and Penn (2010) use the C&C tool `generate` to convert P&K output to the C&C evaluation dependency format. `generate` critically does not depend on the C&C parsing model, permitting a fair comparison of the parsers' output.

	PCTB 5	+PCTB 6	#sents
Train	1–815, 1001–1136	2000–2980	22033
Test	816–885, 1137–1147	3030–3145	2758
Dev	900–931, 1148–1151	2981–3029	1101

Table 1: PCTB 5 and 6 dev/train/test splits

4.1 Evaluation

Carroll et al. (1998) argued against PARSEVAL in favour of a dependency-based evaluation. Rimell et al. (2009) focus on evaluating NLD recovery, proposing a dependency-based evaluation and a GR mapping procedure for inter-parser comparison.

Since the P&K parser plus `generate` produce dependencies in the same format as C&C, we can use the standard Clark and Curran (2007) dependency-based evaluation from the CCG literature: labelled F-score (LF) over dependency tuples, as used for CCG parser evaluation in English. Critically, this metric is also NLD-sensitive. We also report labelled sentence accuracy (Lsa), the proportion of sentences for which the parser returned all and only the gold standard dependencies. Supertagger accuracy compares leaf categories against the gold standard (*stag*).

For C&C, we report on two configurations: GOLD, evaluated using gold standard POS tags; and AUTO, with automatic POS tags provided by the C&C tagger (Curran and Clark, 2003). For P&K, we vary the number of split-merge iterations from one to six (following Fowler and Penn (2010), the k-iterations model is called *I*-k). Because the P&K parser does not use POS tags, the most appropriate comparison is against the AUTO configuration of C&C. For C&C, we use the average of the logarithm of the chart size ($\log C$) as a measure of ambiguity, that is, the number of alternative analyses the parser must choose between.

Following Fowler and Penn (2010), we perform two sets of experiments: one evaluated over all sentences in a section, and another evaluated only over sentences for which both parsers successfully parse and generate dependencies.

We define the *size* of a CCG grammar as the number of categories it contains. The size of a grammar affects the difficulty of the supertagging task (as the size of a grammar is the size of the supertag set). We also consider the number of categories of each *shape*, as defined in Table 2. Decomposing the category in-

Shape	Pattern
V (predicate-like)	$(S[dcl]\backslash NP)\$$
M (modifier)	$X\|X$
P (preposition-like)	$(X\|X)\|Y$
N (noun-like)	N or NP
O (all others)	

Table 2: Shapes of categories

	model	LF	$Lsa\ \%$	stag	cov	$\log C$
A	I-3	68.97	13.45	83.64	95.7	-
	I-6	**71.67**	15.70	85.00	96.4	-
	GOLD	75.45	16.70	89.43	99.4	14.55
	AUTO	66.32	12.81	83.88	98.6	14.69
B	I-3	69.75	14.15	84.07	96.0	-
	I-5	**71.40**	14.83	84.97	96.4	-
	GOLD	75.41	16.67	89.50	99.6	14.74
	AUTO	66.24	12.61	83.95	98.7	14.75
C	I-3	70.22	16.49	84.37	96.5	-
	I-5	**72.74**	18.59	85.61	96.5	-
	GOLD	76.73	20.56	89.66	99.5	13.58
	AUTO	66.95	14.62	83.90	99.2	13.86

Table 3: Dev set evaluation for P&K and C&C

ventory into shapes demonstrates how changes to the corpus annotation affect the distribution of types of category. Finally, we calculate the average number of tags per lexical item (*Avg. Tags/Word*), as a metric of the degree of lexical ambiguity in each corpus.

5 Results

Table 3 shows the performance of P&K and C&C on the three dev sets, and Table 4 only over sentences parsed by both parsers. (**A** is the base release, **B** includes the unary rule $LCP \rightarrow NP$, and **C** also collapses the *N-NP* distinction.) For P&K on corpus **A**, F-score and supertagger accuracy increase monotonically as further split-merge iterations refine the model. P&K on **B** and **C** overfits at 6 iterations, consistent with Fowler and Penn's findings for English.

The $\sim$9% drop in F-score between the GOLD and AUTO figures shows that C&C is highly sensitive to POS tagging accuracy (92.56% on the dev set, compared to 96.82% on English). Considering Table 4, each best P&K model outperforms the corresponding AUTO model by 3-5%. However, while P&K is substantially better without gold-standard information, gold POS tags allow C&C to outperform P&K, again

	model	LF	$Lsa\ \%$	stag	cov
A	I-6	71.74	15.87	85.29	100.0
	AUTO	67.50	15.36	84.52	100.0
B	I-5	71.40	14.97	85.26	100.0
	AUTO	67.72	14.97	84.68	100.0
C	I-5	72.84	18.69	86.04	100.0
	AUTO	68.43	16.17	84.57	100.0

Table 4: Dev set evaluation for P&K and C&C on PCTB 6 sentences parsed by *both* parsers

	model	LF	$Lsa\ \%$	stag	cov	$\log C$
C	I-5	72.73	20.28	85.43	97.1	-
	GOLD	76.89	22.90	89.63	99.1	14.53
	AUTO	67.09	15.28	83.95	98.7	14.89

Table 5: Test set evaluation for P&K and C&C

demonstrating the impact of incorrect POS tags.

Supertagging and parsing accuracy are not entirely correlated between the parsers – in corpora **A** and **B**, AUTO supertagging is comparable or better than I-3, but F-score is substantially worse.

Comparing **A** and **B** in Table 3, C&C receives small increases in supertagger accuracy and coverage, but parsing performance remains largely unchanged; P&K performance degrades slightly. On both parsers, **C** yields the best results out of the three corpora, with LF gains of 1.07 (P&K), 1.28 (GOLD) and 0.63 (AUTO) over the base Chinese CCGbank. We select **C** for our remaining parser experiments.

Both C&C's GOLD and AUTO results show higher coverage than P&K (a combination of parse failures in P&K itself, and in `generate`). Since F-score is only computed over successful parses, it is possible that P&K is avoiding harder sentences. In Table 4, evaluated only over sentences parsed by *both* parsers shows that as expected, C&C gains more (1.15%) than P&K on the common sentences.

Table 5 shows that the behaviour of both parsers on the test section is consistent with the dev section.

corpus	Avg. tags/word	Grammar size all	Grammar size $f \geq 10$
A	1.84	1177	324
B	1.83	1084	303
C	1.79	964	274

Table 6: Corpus statistics

corpus	V	P	M	N	O	*Total*
A	791	158	56	2	170	1177
B	712	149	55	2	166	1084
C	670	119	41	1	133	964

Table 7: Grammar size, categorised by shape

Confusion	LF	ΔLF	stag	cov
Base (GOLD)	76.73		89.66	99.50
NR $\bowtie$ NN	76.72	*-0.01*	89.64	99.37
JJ $\bowtie$ NN	76.60	*-0.12*	89.57	99.37
DEC $\bowtie$ DEG	75.10	*-1.50*	89.07	98.83
VV $\bowtie$ NN	73.35	*-1.75*	87.68	98.74
All (AUTO)	66.95		83.90	99.20

Table 8: Corrupting C&C gold POS tags piecemeal on PCTB 6 dev set of corpus **C**. ΔLF is the change in LF when each additional confusion type is allowed.

5.1 Corpus ambiguity

To understand why corpus **C** is superior for parsing, we compare the ambiguity and sparsity characteristics of the three corpora. Examining $\log C$, the average log-chart size (Table 3) shows that the corpus **B** changes (the addition of the unary rule *LCP* $\rightarrow$ *NP*) increase ambiguity, while the additional corpus **C** changes (eliminating the *N-NP* distinction, resulting in the removal of the unary rule *N* $\rightarrow$ *NP*) have the net effect of reducing ambiguity.

Table 6 shows that the changes reduce the size of the lexicon, thus reducing the average number of tags each word can potentially receive, and therefore the difficulty of the supertagging task. This, in part, contributes to the reduced $\log C$ values in Table 3. While the size of the lexicon is reduced in **B**, the corresponding $\log C$ figure in Table 3 increases slightly, because of the additional unary rule.

Table 7 breaks down the size of each lexicon according to category shape. Introducing the rule *LCP* $\rightarrow$ *NP* reduces the number of **V**-shaped categories by 10%, while not substantially affecting the quantity of other category shapes, because the subcategorisation frames which previously referred to *LCP* are no longer necessary. Eliminating the *N-NP* distinction, however, reduces the number of **P** and **M**-shaped categories by over 20%, as the distinction is no longer made between attachment at *N* and *NP*.

6 Error analysis

The well-known noun/verb ambiguity in Chinese (where, e.g., 设计建设 *'design-build'* is both a verbal compound *'design and build'* and a noun compound *'design and construction'*) greatly affects parsing accuracy (Levy and Manning, 2003).

However, little work has quantified the impact of noun/verb ambiguity on parsing, and for that matter, the impact of other frequent confusion types. To quantify C&C's sensitivity to POS tagging errors, which we saw in Table 3, we perform an experiment where we corrupt the gold POS tags, by gradually reintroducing automatic POS errors on a cumulative basis, one confusion type at a time.

The notation X $\bowtie$ Y indicates that the POS tags X and Y are frequently confused with each other by the POS tagger. For example, VV $\bowtie$ NN represents the problematic noun/verb ambiguity, allowing the inclusion of noun/verb confusion errors.

Table 8 shows that while the confusion types NR $\bowtie$ NN and JJ $\bowtie$ NN have no impact on the evaluation, the confusions DEC $\bowtie$ DEG and VV $\bowtie$ NN, introduced one at a time, cause reductions in F-score of 1.50 and 1.75% respectively. This is expected; Chinese CCGbank does not distinguish between noun modifiers (NN) and adjectives (JJ). On the other hand, the critical noun/verb ambiguity, and the confusion between DEC/DEG (two senses of the particle 的 *de*) adversely impact F-score. We performed an experiment with C&C to merge DEC and DEG into a single tag, but found that this increased category ambiguity without improving accuracy.

The VV $\bowtie$ NN confusion is particularly damaging to the CCG labelled dependency evaluation, because verbs generate a large number of dependencies. While Fowler and Penn (2010) report a gap of 6.31% between C&C's labelled and unlabelled F-score on the development set in English, we observe a gap of 10.35% for Chinese.

Table 10 breaks down the 8,414 false positives generated by C&C on the dev set, according to whether the head of each dependency was incorrectly POS-tagged and/or supertagged. The top-left cell shows that despite the correct POS and supertag, C&C makes a large number of pure attachment location errors. The vast majority of false positives, though, are

C&C AUTO		P&K I-5		category	NLD?	dependency function
LF	freq	*LF*	freq			
0.78	4204	0.78	3106	*NP*/**NP**		*noun modifier attachment*
0.73	2173	0.81	1765	(*S*[*dcl*]*NP*)/**NP**		*transitive object*
0.65	1717	0.72	1459	(*S*[*dcl*]**NP**)/*NP*		*transitive subject*
0.68	870	0.74	643	(*S*[*dcl*]*NP*)/(**S**[**dcl**]**NP**)		*control/raising S complement*
0.70	862	0.67	697	*S*[*dcl*]**NP**		*intransitive subject*
0.60	670	0.69	499	(*S*[*dcl*]**NP**)/(*S*[*dcl*]*NP*)	✓	*control/raising subject*
0.55	626	0.54	412	(*NP*/*NP*)/(**NP**/**NP**)		*noun modifier modifier attachment*
0.57	370	0.68	321	(*NP*/*NP*)\(**S**[**dcl**]**NP**)		*subject extraction S complement*
0.59	343	0.70	314	(*NP*/**NP**)\(*S*[*dcl*]*NP*)	✓	*subject extraction modifier attachment*
0.59	110	0.69	84	(*NP*/*NP*)\(**S**[**dcl**]/**NP**)		*object extraction S complement*
0.63	106	0.75	86	(*NP*/**NP**)\(*S*[*dcl*]/*NP*)	✓	*object extraction modifier attachment*

Table 9: Accuracy per dependency, for selected dependency types

	correct POS		incorrect POS	
correct stag	2307	(27.42%)	51	(0.61%)
incorrect stag	4493	(53.40%)	1563	(18.58%)

Table 10: Analysis of the 8,414 false positive dependencies from C&C on PCTB 6 dev set

caused by supertagging errors (the bottom row), but most of these are not a result of incorrect POS tags, demonstrating that supertagging and parsing are difficult even with correct POS tags.

The sensitivity of C&C to tagging errors, and the higher performance of the P&K parser, which does not directly use POS tags, calls into question whether POS tagging yields a net gain in a language where distinctions such as the noun/verb ambiguity are often difficult to resolve using local tagging approaches. The approach of Auli and Lopez (2011), which achieves superior results in English CCG parsing with a joint supertagging/parsing model, may be promising in light of the performance difference between P&K and C&C.

6.1 Non-local dependencies

Table 9 shows how well the best models of each parser recovered selected local and non-local dependencies. The slot represented by each row appears in boldface. While C&C and P&K perform similarly recovering *NP*-internal structure, the ability of P&K to recover verbal arguments, unbounded long-range dependencies such as subject and object extraction, and bounded long-range dependencies such as control/raising constructions, is superior.

The C&C AUTO parser appears to be biased towards generating far more of the frequent dependency types, yet does not typically have a higher recall for these dependency types than P&K.

6.2 Pro-drop and its impact on CCG parsing

One of the most common types of unary rules in Chinese CCGbank, occurring in 36% of Chinese CCGbank sentences, is the *subject pro-drop rule* $S[dcl]\backslash NP \rightarrow S[dcl]$, which accounts for the optional absence of the subject pronoun of a verb for pragmatic reasons where the referent can be recovered from the discourse (Li and Thompson, 1989).

The subject pro-drop rule is problematic in Chinese parsing because its left hand side, $S[dcl]\backslash NP$, is a very common category, and also because several syntactic distinctions in Chinese CCGbank hinge on the difference between $S[dcl]\backslash NP$ and $S[dcl]$.

The latter point is illustrated by two of the senses of 的 *de*, the Chinese subordinating particle. Two categories which 的 *de* receives in the grammar are $(NP/NP)\backslash(S[dcl]\backslash NP)$ (introducing a relative clause) and $(NP/NP)\backslash S[dcl]$ (in the construction *S de NP*). Because subject pro-drop promotes any unsaturated $S[dcl]\backslash NP$ to $S[dcl]$, whenever the supertagger returns both of the above categories for the lexical item 的 *de*, the parser must consider two alternative analyses which yield different dependencies:

(5) a. t_i 出来 的 问题$_i$
t_i come out DE question$_i$
the questions which arise

	English	Chinese
PTB/PCTB-based	92.1% (McClosky et al., 2006)	86.8% (Zhang and Clark, 2009)
CCGbank-based	86.0% (Fowler and Penn, 2010)	72.7% (this work)
	85.8% (Clark and Curran, 2007)	67.1% (this work)

Table 11: Summary of Chinese parsing approaches

	model	LF	Lsa %	stag	cov	log C
C	GOLD	74.99	7.42	89.36	98.6	18.35
		(76.73	*20.56*	*89.66*	*99.5*	*13.58)*
	AUTO	65.42	4.82	83.73	97.9	18.67
		(66.95	*14.62*	*83.90*	*99.2*	*13.86)*
	I-5	70.67	8.62	84.99	93.8	-
		(72.74	*18.59*	*85.61*	*96.5*	*-)*

Table 12: Dev set evaluation for C&C over *pro*-drop sentences only (and over full set in parentheses)

b. *pro* 出来 的 问题
pro come out DE question
the question of (him, her) coming out

38.1% of sentences in the development set contain at least one instance of *pro*-drop. The evaluation over only these sentences is given in Table 12. This restricted evaluation shows that while we cannot conclude that *pro*-drop is the causative factor, sentences with *pro*-drop are much more difficult for *both* parsers to analyse correctly, although the drops in F-score and supertagging accuracy are largest for P&K.

Critically, the fact that supertagging performance on these more difficult sentences is reasonably comparable with performance on the full set suggests that the bottleneck is in the parser rather than the supertagger. One measure of the complexity of *pro*-drop sentences is the substantial increase in the log C value of these sentences. This suggests that a key to bringing parser performance on Chinese in line with English lies in reining in the ambiguity caused by very productive unary rules such as *pro*-drop.

7 Conclusion

Using Chinese CCGbank (Tse and Curran, 2010), we have trained and evaluated the first CCG parsers for Chinese in the literature: the Clark and Curran (C&C; 2007) and Petrov and Klein (P&K; 2007) parsers. The P&K parser substantially outperformed (72.73) C&C with automatic POS tags (67.09).

Table 11 summarises the best performance of parsers on PTB and CCGbank, for English and Chinese. We observe a drop in performance between English and Chinese CCG parsers which is much larger than, but consistent with, PTB parsers. To close this gap, future research in Chinese parsing should be informed by quantifying the aspects of Chinese which account most for the deficit.

We start by using corpus conversion to compare different linguistic representation choices, rather than for generating a single immutable resource. This can also be exploited to develop syntactic corpora parameterised for particular applications. We found that collapsing categorial distinctions motivated by theory can yield less ambiguous corpora, and hence, more accurate parsers. We have also taken a novel approach to investigating the impact of noun/verb and other POS ambiguities on parsing.

The large gap between Chinese C&C and P&K is surprising, given that Fowler and Penn (2010) found only a small gap for English. We found that C&C is very sensitive to POS tagging performance, which leads to its inferior performance given automatically assigned POS tags. This suggests that joint supertagging/parsing approaches, as performed by P&K, are more suitable for Chinese. Finally, we have shown that *pro*-drop is correlated with poor performance on both parsers, suggesting an avenue to closing the Chinese-English parsing gap.

While developing the first wide-coverage Chinese CCG parsers, we have shed light on the nature of the Chinese-English parsing gap, and identified new and significant challenges for CCG parsing.

Acknowledgements

We thank our anonymous reviewers for their insightful and detailed feedback. James R. Curran was supported by Australian Research Council (ARC) Discovery grant DP1097291 and the Capital Markets Cooperative Research Centre.

References

Michael Auli and Adam Lopez. 2011. A comparison of loopy belief propagation and dual decomposition for integrated ccg supertagging and parsing. In *49th Annual Meeting of the Association for Computational Linguistics*, pages 470–480. Association for Computational Linguistics.

Daniel M. Bikel. 2004. *On the parameter space of generative lexicalized statistical parsing models*. Ph.D. thesis, Citeseer.

Daniel M. Bikel and David Chiang. 2000. Two statistical parsing models applied to the Chinese Treebank. In *Second workshop on Chinese language processing*, volume 12, pages 1–6. Morristown, NJ, USA.

John Carroll, Ted Briscoe, and Antonio Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. In *Proceedings of the 1st International Conference on Language Resources and Evaluation*, pages 447–454.

Yuen-Ren Chao. 1968. *A grammar of spoken Chinese*. University of California Press.

David Chiang and Daniel M. Bikel. 2002. Recovering latent information in treebanks. In *Proceedings of the 19th international conference on Computational linguistics*, volume 1, pages 1–7. Association for Computational Linguistics.

Stephen Clark and James R. Curran. 2004. The importance of supertagging for wide-coverage CCG parsing. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics.

Stephen Clark and James R. Curran. 2007. Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models. In *Computational Linguistics*, volume 33, pages 493–552.

James R. Curran and Stephen Clark. 2003. Investigating GIS and smoothing for maximum entropy taggers. In *Proceedings of the 10th Meeting of the EACL*, pages 91–98. Budapest, Hungary.

Timothy A.D. Fowler and Gerald Penn. 2010. Accurate context-free parsing with combinatory categorial grammar. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 335–344.

Yuqing Guo, Haifeng Wang, and Josef Van Genabith. 2007. Recovering non-local dependencies for Chinese. In *EMNLP/CoNLL*, pages 257–266.

Julia Hockenmaier. 2003. *Data and Models for Statistical Parsing with Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh.

Julia Hockenmaier and Mark Steedman. 2005. CCGbank: Users' manual. Technical report, MS-CIS-05-09, Computer and Information Science, University of Pennsylvania.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: A Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Matthew Honnibal. 2010. *Hat Categories: Representing Form and Function Simultaneously in Combinatory Categorial Grammar*. Ph.D. thesis, University of Sydney.

Liang Huang, Wenbin Jiang, and Qun Liu. 2009. Bilingually-constrained (monolingual) shift-reduce parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, volume 3, pages 1222–1231. Association for Computational Linguistics.

Roger Levy and Christopher Manning. 2003. Is it harder to parse Chinese, or the Chinese Treebank? In *Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 439–446. Morristown, NJ, USA.

Charles N. Li and Sandra A. Thompson. 1989. *Mandarin Chinese: A functional reference grammar*. University of California Press.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1994. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 152–159. Association for Computational Linguistics.

Yusuke Miyao, Takashi Ninomiya, and Jun'ichi Tsujii. 2004. Corpus-Oriented Grammar Development for Acquiring a Head-Driven Phrase Structure Grammar from the Penn Treebank. pages 684–693.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of NAACL HLT 2007*, pages 404–411.

Laura Rimell, Stephen Clark, and Mark Steedman. 2009. Unbounded dependency recovery for parser evaluation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 813–821. Association for Computational Linguistics.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press. Cambridge, MA, USA.

Daniel Tse and James R. Curran. 2010. Chinese CCGbank: extracting CCG derivations from the Penn Chinese Treebank. *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, pages 1083–1091.

Mengqiu Wang, Kenji Sagae, and Teruko Mitamura. 2006. A fast, accurate deterministic parser for Chinese. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 425–432. Association for Computational Linguistics.

Fei Xia. 1999. Extracting tree adjoining grammars from bracketed corpora. In *Proceedings of Natural Language Processing Pacific Rim Symposium '99*, pages 398–403.

Fei Xia, Chung-hye Han, Martha Palmer, and Aravind Joshi. 2000. Comparing lexicalized treebank grammars extracted from Chinese, Korean, and English corpora. In *Proceedings of the second workshop on Chinese language processing: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics*, volume 12, pages 52–59. Association for Computational Linguistics.

Nianwen Xue, Fei Xia, Fu-Dong Chiou, and Martha Palmer. 2005. The Penn Chinese TreeBank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(02):207–238.

Kun Yu, Yusuke Miyao, Takuya Matsuzaki, Xiangli Wang, and Jun'ichi Tsujii. 2011. Analysis of the difficulties in chinese deep parsing. In *12th International Conference on Parsing Technologies*, page 48.

Yue Zhang and Stephen Clark. 2008. A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 562–571. Association for Computational Linguistics.

Yue Zhang and Stephen Clark. 2009. Transition-based parsing of the Chinese treebank using a global discriminative model. In *Proceedings of the 11th International Conference on Parsing Technologies*, pages 162–171. Association for Computational Linguistics.