COPYRIGHT AND USE OF THIS THESIS

This thesis must be used in accordance with the provisions of the Copyright Act 1968.

Reproduction of material protected by copyright may be an infringement of copyright and copyright owners may be entitled to take legal action against persons who infringe their copyright.

Section 51 (2) of the Copyright Act permits an authorized officer of a university library or archives to provide a copy (by communication or otherwise) of an unpublished thesis kept in the library or archives, to a person who satisfies the authorized officer that he or she requires the reproduction for the purposes of research or study.

The Copyright Act grants the creator of a work a number of moral rights, specifically the right of attribution, the right against false attribution and the right of integrity.

You may infringe the author’s moral rights if you:

- fail to acknowledge the author of this thesis if you quote sections from the work
- attribute this thesis to another author
- subject this thesis to derogatory treatment which may prejudice the author’s reputation

For further information contact the University’s Director of Copyright Services

sydney.edu.au/copyright
CHINESE CCGBANK: DEEP DERIVATIONS
AND DEPENDENCIES
FOR CHINESE CCG PARSING

Daniel Gar-Shon Tse

Supervisor: Dr. James Curran

THE UNIVERSITY OF SYDNEY

2013
Abstract

Parsing, the act of retrieving the latent structure from the linear medium of words, is necessary in many applications which aim to understand natural language as humans do; without this structure, text is merely a sequence of tokens.

The fundamental goal of this dissertation is to establish that deep, efficient, accurate parsing models can be acquired for Chinese, through parsers founded on Combinatory Categorial Grammar (ccg; Steedman, 2000), a grammar formalism which has already enabled the creation of rich parsing models for English.

ccg boasts succinct type-driven accounts of extraction, passivisation and other complex phenomena, as well as efficient parsers capable of recovering these analyses, including the state-of-the-art ccg parser C&C (Clark and Curran, 2007b).

We harness these ccg analyses of cross-linguistic syntax, harmonising them with modern accounts from Chinese generative syntax, contributing the first analysis of Chinese syntax through ccg in the literature. Our abstract analysis captures a host of familiar constructions from Chinese syntax such as the 把/被 ba/bei construction, topicalisation and the null copula construction.

Supervised statistical parsing approaches rely on the availability of large annotated corpora. To avoid the cost of manual annotation, we adopt the corpus conversion methodology, in which an automatic corpus conversion algorithm projects annotations from a source corpus into the target formalism. Corpus conversion has previously been used to induce wide-coverage HPSG (Miyao et al., 2004), LFG (Cahill et al., 2002; Guo et al., 2007a), LTAG (Xia et al., 2000a) and ccg grammars (Hockenmaier, 2003, 2006).

We develop a corpus conversion which extracts the abstract ccg analysis of Chinese syntax from Penn Chinese Treebank trees, demonstrating that our abstract analyses of a host of Chinese syntactic constructions including topicalisation, extraction, passivisation and pro-drop can be extracted automatically from the Penn Chinese Treebank.

The central contribution of this thesis is Chinese CCGBank, a corpus of 750,000 words automatically extracted from the Penn Chinese Treebank, reifying the abstract analysis through corpus conversion.

We then take three state-of-the-art ccg parsers from the literature — the split-merge PCFG parser of Petrov and Klein (P&K; 2007), the transition-based ccg parser of Zhang et al. (Z&C; 2011), and the maximum entropy parser of Clark and Curran (C&C; 2007b) — and train and evaluate all three on Chinese CCGBank, achieving the first Chinese ccg parsing models in the literature.
We demonstrate that while the three parsers are only separated by a small margin trained on English CCGbank, a substantial gulf of 4.8% separates the same parsers trained on Chinese CCGbank. We also confirm that the gap between the states-of-the-art in English and Chinese PSG parsing can be observed in CCG parsing.

First, we develop the methodology of parser-aided corpus conversion, which empirically evaluates corpus conversion decisions by training the resulting corpus on a wide-coverage parser. This allows us to quantify the impact of our annotation decisions on the performance of parsing models trained on the resulting corpus. Then, spurred by early observations by Levy and Manning (2003) that the noun/verb ambiguity in Chinese leads to considerable parser error, we investigate the role of tagger error in Chinese parsing performance.

Our parsing experiments establish Chinese CCG parsing as a new and substantial challenge, a line of empirical investigation directly enabled by Chinese CCGbank.
Acknowledgements

This dissertation is the product of many kindnesses. May these words share my happiness with those who helped me write it.

My parents, Pandora and Peter, nurtured a love of language which has served me well and steered the rudder of my life. They must have wanted me to become a computer scientist; well before there was a computer in every home, they ensured there were many in ours. Affection and gratitude go to all my family, especially my brother Edwin (a scientist in training).

One day, a prospective undergraduate found his way to an office in the Basser Department of Computer Science, an old building for a modern science. That office belonged to my mentor and supervisor, Dr. James Curran. He taught me how to do science and how to communicate science, always with warmth and concern, always with a sense of humour, always with joy and enthusiasm. As I teach the generation to come, I will strive to do the same.

I thank Prof. Jane Simpson, my associate supervisor 2008–2010. As computational linguistics benefits from its ties to linguistics, so has this thesis benefited from your advice.

I have seen the research group at Sydney enriched by the enthusiasm, knowledge and work of many keen students. I am grateful to be a part of θ-lab,¹ and grateful to my colleagues for their support and feedback. My colleagues and friends at the lab — Will Radford, Tim

¹ Schwa Lab (http://schwa.org): a most central place, a most central vowel.
O’Keefe, Joel Nothman, Dominick Ng, James Constable, Tim Dawborn, Nicky Ringland, Andrew Naoum, Glen Pink, Richard Billingsley — have enriched my time at Sydney with the spirit of their research, feedback on my writing and work, numerous lab lunches, and the occasional lab retreat or conference. Thanks also to Candice Loxley and the duty programmers for supporting my research.

In 2011, I spent five summer months at Microsoft Research Asia as a Microsoft Fellowship recipient, making balmy Beijing my next academic home. My supervisor Professor Ming Zhou (周明) and researcher Xiaohua Liu (刘小华) were enthusiastic supporters of my work, and imparted valuable experience working in a large commercial research lab.

I would like to thank in particular Professors Mark Steedman and Julia Hockenmaier for their insightful advice on ccg, CCGbank and the creation of wide-coverage corpora. I hope that they will recognise their ideas and counsel embodied in this work. I am also grateful to the review panel for their close reading of my work.

Finally, I dedicate this dissertation to my friends, enthusiastic travellers who would follow me to the ends of the earth if a conference were to be held there. With them, many an afternoon and evening was spent in coffee-shops and through Sydney streets, discussing all things, especially the idiosyncrasies of languages (both artificial and natural). Thanks in particular to my good friends Mitchell Quille and Rob Little for being a part of the journey. You mean a lot to me.

There are still so many things in this world to learn. To study is to row a boat against the current — not to advance is to fall behind.

Daniel Gar-Shon Tse
Sydney, Australia
2013
学如逆水行舟不进则退
Statement of compliance

I certify that:

- I have read and understood the University of Sydney Student Plagiarism: Coursework Policy and Procedure;

- I understand that failure to comply with the Student Plagiarism: Coursework Policy and Procedure can lead to the University commencing proceedings against me for potential student misconduct under Chapter 8 of the University of Sydney By-Law 1999 (as amended);

- this Work is substantially my own, and to the extent that any part of this Work is not my own I have indicated that it is not my own by Acknowledging the Source of that part or those parts of the Work.

Name: Daniel Gar-Shon Tse

Signature: Date: 11th October 2013
Contents

0 Introduction ............................ 3
  0.1 Why parse? .......................... 4
  0.2 The thesis .......................... 4
  0.3 The structure of this work ........... 7
  0.4 Notation ............................ 8
  0.5 References to the Penn Chinese Treebank .......................... 9
  0.6 Acquiring a copy of Chinese CCGbank .......................... 9
  0.7 Publications .......................... 9

1 Combinatory Categorial Grammar ............................. 11
  1.1 A gentle introduction to CCG through categorial grammar .... 12
  1.2 The development of grammar formalisms ......................... 17
  1.3 The construction of categories ................................ 19
  1.4 How we depict derivations ................................ 21
  1.5 Decorations on categories ................................ 22
    1.5.1 Features ................................ 22
    1.5.2 Categories as attribute-value matrices .................... 24
    1.5.3 AVM encoding of features ............................... 25
    1.5.4 AVM encoding of head variables ........................... 27
  1.6 Combinatory rules ................................ 28
    1.6.1 Application (>,<) ................................ 28
    1.6.2 The rebracketing puzzle ............................... 29
    1.6.3 Composition (B) ................................ 31
    1.6.4 Type-raising (T) ................................ 33
    1.6.5 Substitution (S) ................................ 34
    1.6.6 Stipulations on combinatory rules ......................... 35
    1.6.7 Stipulations on the lexicon .............................. 36
CONTENTS

1.7 Multi-modal CCG .................................. 37
1.8 Extra-combinatory rules .......................... 39
  1.8.1 Rules unconstrained by the stipulations on CCG rules .... 39
  1.8.2 Absorption rules ............................. 40
1.9 The trade-off between derivational and lexical ambiguity .... 41
  1.9.1 Analyses through lexical ambiguity ...................... 42
  1.9.2 Analyses through derivational ambiguity ................ 43
  1.9.3 Hat categories ................................ 44
1.10 Notational conveniences for categories .................... 45
1.11 Applications of CCG .................................. 46
  1.11.1 Generation and realisation ......................... 46
  1.11.2 Supertagging .................................. 47
  1.11.3 Question answering ............................. 47
1.12 Summary ........................................... 48

2 The elements of Chinese CCG .......................... 49
  2.1 A review of CCG grammars in the literature ................. 50
  2.2 On developing the Chinese CCGbank grammar ............... 52
  2.3 Basic sentence typology and position of adjuncts .......... 54
    2.3.1 Canonical word order ........................ 54
    2.3.2 Subcategorisation in Chinese .................... 56
  2.4 Constructions headed by NP .......................... 57
    2.4.1 Quantifier phrases and measure words ............... 57
    2.4.2 Bare versus non-bare nouns ...................... 61
    2.4.3 Adjective-noun order, and adjectives versus stative verbs .... 64
    2.4.4 NP NP apposition ................................ 66
    2.4.5 S NP apposition ................................. 67
  2.5 Constructions involving the verb .......................... 69
    2.5.1 Aspect particles and other verbal modifiers .......... 69
    2.5.2 V-O forms .................................... 71
  2.6 Coordination phenomena .................................. 73
    2.6.1 The syntax of coordination ........................ 74
    2.6.2 List commas ..................................... 77
    2.6.3 Right node raising .............................. 78
    2.6.4 Argument cluster coordination .................... 79
2.6.5 Unlike coordination phrases .................................. 80
2.7 Sentence-level phenomena .................................. 82
  2.7.1 Yes-no questions .................................. 82
  2.7.2 Direct quoted speech .................................. 83
2.8 Punctuation .................................. 85
  2.8.1 Clause-separating commas .................................. 86
  2.8.2 Sentence modifiers of sentences .................................. 86
  2.8.3 Parentheticals .................................. 87
2.9 Verb compounding strategies .................................. 87
  2.9.1 Verb potential construction (VPT) .................................. 87
  2.9.2 The V-bu-V construction (VNV) .................................. 89
  2.9.3 Verb resultative complements (VRD) .................................. 90
  2.9.4 Verb coordination (VCD) .................................. 91
  2.9.5 Verb subordination (VSB) .................................. 93
2.10 Summary .................................. 95

3 The analysis of special syntax .................................. 97
  3.1 Control verbs and modal verbs .................................. 97
  3.2 Relative clauses and extraction .................................. 99
    3.2.1 Gapless versus gapped 的 de constructions .................................. 100
    3.2.2 Overt versus null relativiser .................................. 101
    3.2.3 Generalised 的 de modification .................................. 102
    3.2.4 Pseudo-cleft construction .................................. 103
    3.2.5 On extraction islands .................................. 106
  3.3 Topicalisation and topic-prominence .................................. 108
  3.4 把 ba and 被 bei constructions .................................. 111
    3.4.1 The long 被 bei construction .................................. 113
    3.4.2 The short 被 bei construction .................................. 118
    3.4.3 The 把 ba construction .................................. 121
    3.4.4 Summary of the 把/被 ba/bei constructions .................................. 122
  3.5 pro-drop .................................. 123
    3.5.1 Three approaches to modelling pro-drop .................................. 126
  3.6 Null copula .................................. 128
  3.7 wh-questions .................................. 129
    3.7.1 The 为什么 why/怎么样 how asymmetry .................................. 133
CONTENTS

3.8 Summary .............................................................. 136

4 Creating Chinese CCGbank ................................. 139
  4.1 The Penn Chinese Treebank ............................... 140
    4.1.1 Differences between PTB and PCTB .................. 143
    4.1.2 Dataset splits ........................................ 150
  4.2 CCGbank ......................................................... 151
  4.3 Comparisons with other conversion algorithms ............ 152
    4.3.1 English CCGbank ...................................... 152
    4.3.2 CCGbanks from dependency corpora: TUT-CCG Italian CCGbank and a German CCGbank ..... 152
  4.4 Summary of the conversion algorithm ....................... 153
  4.5 The generation algorithm ...................................... 153
  4.6 The Chinese CCGbank conversion algorithm ............... 156
    4.6.1 PREPROCESS(T) ........................................ 156
    4.6.2 MARK(T) ................................................ 161
    4.6.3 BINARISE(T) ........................................... 162
    4.6.4 CATLAB(T) ............................................. 165
    4.6.5 FIX(T) ................................................ 167
  4.7 Summary .......................................................... 173

5 Extracting NLD analyses for Chinese CCGbank ...... 175
  5.1 Traces ............................................................. 176
    5.1.1 Describing transformation procedures ................ 177
    5.1.2 pro traces ............................................. 179
    5.1.3 PRO traces ............................................. 180
    5.1.4 Traces of wh-movement ................................ 183
    5.1.5 Right node raising ..................................... 191
    5.1.6 Argument cluster coordination ...................... 195
    5.1.7 Instances of the 把 ba and 被 bei constructions .......... 197
  5.2 Constructions not handled .................................... 204
  5.3 Gold-standard dependency extraction ..................... 208
  5.4 Evaluating automatic gold-standard NLD extraction ....... 209
  5.5 Analysis of Chinese CCGbank ............................... 213
    5.5.1 The lexicon ........................................... 213
    5.5.2 Coverage losses ....................................... 215
CONTENTS

5.5.3 The effect of sparsity ................................................. 216
5.5.4 Discharging NLD traces ............................................. 216
5.6 Summary ................................................................. 219

6 Parsing Chinese CCGbank .............................................. 221
  6.1 A roadmap ............................................................... 223
  6.2 A survey of approaches to Chinese statistical parsing ......... 224
  6.3 Approaches to CCG parsing ......................................... 226
    6.3.1 C&C: log-linear models for efficient CCG parsing .......... 227
    6.3.2 P&K: PCFG-based CCG parsing ................................ 231
    6.3.3 Z&C: Transition-based CCG parsing ......................... 233
  6.4 Components common to all three parsers ......................... 236
    6.4.1 The markedup file ............................................... 236
    6.4.2 One structural category maps to multiple markedup entries ... 237
    6.4.3 Format of the markedup file .................................... 238
    6.4.4 Generating the markedup file automatically ................. 239
    6.4.5 Evaluating the automatic markedup file generation algorithm .... 240
    6.4.6 generate tool .................................................... 243
    6.4.7 CCG dependency-based evaluation .............................. 244
  6.5 Chinese POS tagging ............................................... 246
    6.5.1 Evaluating modifications to the POS tagger ................ 247
    6.5.2 Prosodic features for POS tagging ............................ 248
  6.6 Chinese supertagging ................................................ 250
  6.7 Models for Chinese parsing ......................................... 251
  6.8 Modifying the C&C parser for Chinese parsing .................. 254
  6.9 Notes on the experimental setup ................................... 255
  6.10 Multi-parser empirical evaluation of Chinese CCGbank ......... 256
  6.11 Analysis of parsing results ........................................ 260
  6.12 Summary ............................................................... 261

7 The challenges of parsing Chinese cccg ................................ 263
  7.1 The parser-aided corpus conversion methodology ............... 264
    7.1.1 Three versions of Chinese CCGbank ............................ 265
    7.1.2 Evaluating static properties of the corpus .................. 267
    7.1.3 Results .......................................................... 267
    7.1.4 Corpus ambiguity ............................................... 269
## CONTENTS

7.1.5 Summary .......................................................... 270
7.2 Coverage loss behaviour for each parser ....................... 272
7.3 Corrupting pos tags to isolate confusions ..................... 274
7.4 The effect of supertagger accuracy ............................. 276
7.5 Analysis of parser accuracy by dependency type ............ 276
7.6 pro-drop and its impact on ccg parsing ....................... 278
7.7 Summary ............................................................... 280

8 Conclusion .................................................................. 283
  8.1 Looking back ......................................................... 283
  8.2 Looking forward ..................................................... 285
  8.3 Conclusion ............................................................ 286

A Unary projections in the Penn Chinese Treebank ............... 287

B Type-change rules in Chinese CCGbank ......................... 289

C Fixes to РCTВ annotation .......................................... 291

Bibliography ................................................................ 297
## List of Figures

1 A roadmap of this thesis ................................................................. 7
1.1 Attribute-value matrix representation of \( \theta_1 \) ............................ 25
1.2 The innermost categories of the argument and result categories have a coin-
dexed feature variable ................................................................. 26
1.3 Harmonic composition licenses rebracketing ................................. 32
1.4 Crossed composition licenses reordering ....................................... 32
1.5 Type-raising leads to multiple “spurious” derivations ....................... 33
1.6 Simplified mode hierarchy .......................................................... 38

4.1 A Penn Chinese Treebank derivation (glosses provided by us) .......... 141
4.2 Composition of PCTB 6.0 ............................................................ 142
4.3 HCA relation is implicit in PCTB structure .................................. 155
4.4 Marking makes HCA relations explicit ......................................... 155
4.5 Head and its complements are siblings ......................................... 155
4.6 Binarisation must reflect head structure ....................................... 156
4.7 PCTB segmentation for infixing morphology ................................ 158
4.8 Re-analysis of serial verb construction ....................................... 160
4.9 Typical unary projections ........................................................... 162
4.10 Intermediate trees may require category repair ............................. 169

5.1 In LFG, traces in c-structure correspond to re-entrancies in f-structure .. 177
5.2 IP with controlled PRO subject corresponds to \( S[del] \backslash NP \) ........ 184
5.3 Chinese CCGbank analysis of gapped topicalisation ....................... 190
5.4 Category growth behaviour on the Penn Chinese Treebank ............. 217

6.1 The impact of \( \beta \) on multitagger ambiguity ............................... 229
6.2 Architecture of the C&C parser .................................................... 230
6.3 Rule threshold \( \tau \) against coverage (blue series) and F-score (purple series) . 235
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.4</td>
<td>Markedup entry for category ((S[dcl]\NP_y)/(S[dcl]<em>z\NP_w)</em>{z}/\NP_{y})_{z})</td>
<td>239</td>
</tr>
<tr>
<td>6.5</td>
<td>Performance by dependency length</td>
<td>258</td>
</tr>
<tr>
<td>6.6</td>
<td>(P) and (R) by dependency length</td>
<td>259</td>
</tr>
<tr>
<td>B.1</td>
<td>Unary type-change rules in Chinese CCGbank</td>
<td>289</td>
</tr>
<tr>
<td>B.2</td>
<td>Binary type-change rules in Chinese CCGbank</td>
<td>290</td>
</tr>
<tr>
<td>B.3</td>
<td>Licensed type-raising operations in the Chinese CCGbank lexicon</td>
<td>290</td>
</tr>
</tbody>
</table>
List of Tables

1   Orthographic conventions ........................................... 8
1.1  English CCGbank atoms ........................................... 20
1.2  An AVM model of category structure .............................. 24
2.1  Language fragments analysed through CCG ....................... 52
2.2  Basic verbal categories in Chinese and English ................... 55
2.3  Subcategorisation frames in the PCTB, truncated at frequency 100 .... 56
2.4  Partitioning adjectives into predicative, pure or mixed ........... 65
2.5  Highest frequency head nouns in S-NP apposition in PCTB ........ 68
2.6  Chinese aspect markers attested in PCTB by frequency .......... 69
2.7  Chinese coordination words attested in PCTB by frequency ....... 74
2.8  Chinese CCGbank candidate absorption categories ................ 85
2.9  Verb compounding strategies in the Penn Chinese Treebank by frequency in PCTB ........................................... 88
2.10 Common compound-initial and -final elements and their frequency in Penn Chinese Treebank .................................... 94
3.1  Configurations and prevalence of the RC construction in Chinese ..................... 100
3.2  Types of generalised 的 de modification in PCTB down to frequency 100 104
3.3  Prevalence in PCTB of topicalisation constructions .................. 110
3.4  Prevalence in PCTB of 把/被 ba/bei constructions ................... 113
3.5  Category of 被 bei in the CCG analogues of the three analyses of long 被 bei 119
3.6  Summary of 把/被 ba/bei categories .................................. 122
3.7  Frequency of pro-drop in PCTB ...................................... 123
3.8  Most frequently pro-dropped verbs in PCTB down to frequency 99. (% pro) shows the relative frequency of pro-drop for each verb to all occurrences. 124
3.9  Distribution of non-VP predicate types PCTB down to threshold 100 .... 128
<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Comparison of corpora</td>
<td>142</td>
</tr>
<tr>
<td>4.2</td>
<td>Comparison of structural characteristics</td>
<td>142</td>
</tr>
<tr>
<td>4.3</td>
<td>The Penn Chinese Treebank word-level tagset</td>
<td>144</td>
</tr>
<tr>
<td>4.4</td>
<td>The Penn Chinese Treebank phrase-level tagset</td>
<td>145</td>
</tr>
<tr>
<td>4.5</td>
<td>Configurations of rules, with word-level tag H and phrase-level tags XP, YP, HP</td>
<td>150</td>
</tr>
<tr>
<td>4.6</td>
<td>PCTB 5 and 6 dev/train/test splits</td>
<td>151</td>
</tr>
<tr>
<td>4.7</td>
<td>Stages of the generation algorithm</td>
<td>153</td>
</tr>
<tr>
<td>4.8</td>
<td>Markers added in the marking algorithm</td>
<td>162</td>
</tr>
<tr>
<td>4.9</td>
<td>Paired punctuation</td>
<td>165</td>
</tr>
<tr>
<td>4.10</td>
<td>Root mapping for Chinese CCGbank</td>
<td>166</td>
</tr>
<tr>
<td>4.11</td>
<td>Category labelling schemata</td>
<td>167</td>
</tr>
<tr>
<td>4.12</td>
<td>Ordering of trace elimination procedures</td>
<td>170</td>
</tr>
<tr>
<td>5.1</td>
<td>Trace types in the Penn Chinese Treebank annotation</td>
<td>176</td>
</tr>
<tr>
<td>5.2</td>
<td>Graphical presentation of MUNGE expressions</td>
<td>179</td>
</tr>
<tr>
<td>5.3</td>
<td>Types of subjects of non-finite IPs in PCTB</td>
<td>181</td>
</tr>
<tr>
<td>5.4</td>
<td>Types of extraction in Chinese Penn Treebank</td>
<td>189</td>
</tr>
<tr>
<td>5.5</td>
<td>Three configurations of 被 bei in PCTB and one of 把 ba</td>
<td>198</td>
</tr>
<tr>
<td>5.6</td>
<td>NLD types for our manual gold-standard annotation</td>
<td>211</td>
</tr>
<tr>
<td>5.7</td>
<td>Results of manual gold-standard NLD annotation</td>
<td>212</td>
</tr>
<tr>
<td>5.8</td>
<td>Error analysis of 56 mkdeps errors, categorised</td>
<td>212</td>
</tr>
<tr>
<td>5.9</td>
<td>Most ambiguous lexical items in Chinese CCGbank</td>
<td>214</td>
</tr>
<tr>
<td>5.10</td>
<td>Error analysis of coverage losses</td>
<td>215</td>
</tr>
<tr>
<td>5.11</td>
<td>Average lexical and category coverage across CCGbanks (ten-fold cross-validation)</td>
<td>218</td>
</tr>
<tr>
<td>5.12</td>
<td>Traces discharged by the conversion algorithm, by NLD type</td>
<td>219</td>
</tr>
<tr>
<td>6.1</td>
<td>Parser models in Clark and Curran (2007b)</td>
<td>230</td>
</tr>
<tr>
<td>6.2</td>
<td>C&amp;C parameters (chosen values in parentheses)</td>
<td>232</td>
</tr>
<tr>
<td>6.3</td>
<td>Markedup categories for NLDs and exceptional cases in Chinese CCGbank</td>
<td>241</td>
</tr>
<tr>
<td>6.4</td>
<td>Breakdown of 145 cases in which automatic markedup annotations differ</td>
<td>243</td>
</tr>
<tr>
<td>6.5</td>
<td>Results of pos tagger trained with prosodic and reduplicative features</td>
<td>250</td>
</tr>
<tr>
<td>6.6</td>
<td>The effect of extended horizon features on supertagging: two models evaluated on the PCTB 6.0 development set</td>
<td>250</td>
</tr>
<tr>
<td>6.7</td>
<td>Feature functions</td>
<td>251</td>
</tr>
</tbody>
</table>
LIST OF TABLES

6.8 C&C evaluation of extra dependency features on Chinese CCGbank dev
    set, AUTO POS tags ........................................... 253
6.9 C&C parser features ............................................. 253
6.10 Wall-clock time estimates for training and parsing .................. 255
6.11 Evaluation only on sentences parsed by C&C and each other parser ... 256
6.12 Empirical evaluation of Chinese CCGbank on three parsers (dev set) . 257
6.13 Empirical evaluation of Chinese CCGbank on three parsers (test set) . 257
6.14 Summary of Chinese parsing approaches .......................... 260

7.1 Shapes of categories ............................................. 267
7.2 Dev set evaluation for P&K and C&C on the three corpus variations ... 268
7.3 Dev set evaluation for P&K and C&C on рсуб 6 sentences parsed by both
    parsers ............................................................ 268
7.4 Test set evaluation for P&K and C&C on corpus No-Bare-N ............ 269
7.5 Corpus statistics ................................................. 270
7.6 Grammar size, categorised by shape ................................ 270
7.7 Sources of coverage loss in each parser ........................... 271
7.8 Coverage losses for each parser (number of sentences) ............... 274
7.9 Corrupting C&C gold POS tags piecemeal on dev set of Chinese CCGbank.
    ∆LF is the change in LF when each additional confusion type is allowed. 275
7.10 Analysis of the 8511 false positive dependencies from C&C on рсуб 6 dev
    set ................................................................. 276
7.11 Accuracy per dependency, for selected dependency types .............. 279
7.12 Dev set evaluation for C&C and P&K over pro-drop sentences only (and
    over full set in parentheses) ................................... 280

A.1 Unary projections .................................................. 287
## List of Algorithms

<table>
<thead>
<tr>
<th></th>
<th>Algorithm Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The Chinese CCGbank conversion pipeline</td>
<td>157</td>
</tr>
<tr>
<td>2</td>
<td>\texttt{PREPROCESS}(T)</td>
<td>157</td>
</tr>
<tr>
<td>3</td>
<td>Marking algorithm, \texttt{MARK}(T)</td>
<td>163</td>
</tr>
<tr>
<td>4</td>
<td>Category labelling algorithm, \texttt{CATLAB}(node)</td>
<td>166</td>
</tr>
<tr>
<td>5</td>
<td>Trace elimination algorithm, \texttt{FIX}(T)</td>
<td>167</td>
</tr>
<tr>
<td>6</td>
<td>Category repair algorithm \texttt{FIXCAT}(node, until)</td>
<td>171</td>
</tr>
<tr>
<td>7</td>
<td>Generalising over modifier categories</td>
<td>173</td>
</tr>
<tr>
<td>8</td>
<td>Re-analysis of subject extraction</td>
<td>186</td>
</tr>
<tr>
<td>9</td>
<td>Re-analysis of object extraction</td>
<td>187</td>
</tr>
<tr>
<td>10</td>
<td>Re-analysis of gapped topicalisation</td>
<td>191</td>
</tr>
<tr>
<td>11</td>
<td>Re-analysis of argument cluster coordination</td>
<td>196</td>
</tr>
<tr>
<td>12</td>
<td>Re-analysis of subject-gapped short 被 bei</td>
<td>200</td>
</tr>
<tr>
<td>13</td>
<td>Re-analysis of object-gapped short 被 bei</td>
<td>200</td>
</tr>
<tr>
<td>14</td>
<td>Re-analysis of object-gapped long 被 bei</td>
<td>202</td>
</tr>
<tr>
<td>15</td>
<td>Re-analysis of object-gapped 把 ba</td>
<td>204</td>
</tr>
<tr>
<td>16</td>
<td>Distinguishing two markedup entries using a category annotation</td>
<td>238</td>
</tr>
<tr>
<td>17</td>
<td>Markedup generation algorithm, \texttt{MARKMARKED}</td>
<td>240</td>
</tr>
<tr>
<td>18</td>
<td>The stratified shuffling test</td>
<td>247</td>
</tr>
</tbody>
</table>
List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>Ajdukiewicz-Bar Hillel categorial grammar.</td>
</tr>
<tr>
<td>ASP</td>
<td>Aspect particle.</td>
</tr>
<tr>
<td>ASP</td>
<td>Across-the-board condition.</td>
</tr>
<tr>
<td>CCG</td>
<td>Combinatory Categorial Grammar.</td>
</tr>
<tr>
<td>CFG</td>
<td>Context-free grammar.</td>
</tr>
<tr>
<td>CG</td>
<td>Categorial grammar.</td>
</tr>
<tr>
<td>CKY</td>
<td>Cocke-Kasami-Younger algorithm.</td>
</tr>
<tr>
<td>COP</td>
<td>Copula.</td>
</tr>
<tr>
<td>CSC</td>
<td>Coordinate Structure Constraint.</td>
</tr>
<tr>
<td>DC</td>
<td>Directional complement.</td>
</tr>
<tr>
<td>DUR</td>
<td>Durative.</td>
</tr>
<tr>
<td>GB</td>
<td>Government and Binding.</td>
</tr>
<tr>
<td>GCR</td>
<td>Generalised Control Rule.</td>
</tr>
<tr>
<td>GR</td>
<td>Grammatical relation.</td>
</tr>
<tr>
<td>HPSG</td>
<td>Head-driven phrase structure grammar.</td>
</tr>
<tr>
<td>LCM</td>
<td>List comma.</td>
</tr>
<tr>
<td>LC</td>
<td>Localiser.</td>
</tr>
<tr>
<td>LFG</td>
<td>Lexical functional grammar.</td>
</tr>
<tr>
<td>LF</td>
<td>Logical form.</td>
</tr>
<tr>
<td>LTAG</td>
<td>Lexicalised tree adjoining grammar.</td>
</tr>
<tr>
<td>MMCCG</td>
<td>Multi-modal combinatory categorial grammar.</td>
</tr>
<tr>
<td>MW</td>
<td>Measure word.</td>
</tr>
<tr>
<td>PCFG</td>
<td>Probabilistic context-free grammar.</td>
</tr>
<tr>
<td>PCTB</td>
<td>Penn Chinese Treebank.</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>PF</td>
<td>Phonological form.</td>
</tr>
<tr>
<td>POS</td>
<td>Part of speech.</td>
</tr>
<tr>
<td>PTB</td>
<td>Penn Treebank.</td>
</tr>
<tr>
<td>RC</td>
<td>Relative clause.</td>
</tr>
<tr>
<td>RES</td>
<td>Resultative complement.</td>
</tr>
<tr>
<td>RNR</td>
<td>Right node raising.</td>
</tr>
<tr>
<td>RRC</td>
<td>Reduced relative clause.</td>
</tr>
<tr>
<td>SFP</td>
<td>Sentence-final particle.</td>
</tr>
<tr>
<td>SOV</td>
<td>Subject-object-verb word order.</td>
</tr>
<tr>
<td>SRL</td>
<td>Semantic role labelling.</td>
</tr>
<tr>
<td>SVC</td>
<td>Serial verb construction.</td>
</tr>
<tr>
<td>SVO</td>
<td>Subject-verb-object word order.</td>
</tr>
<tr>
<td>TAG</td>
<td>Tree adjoining grammar.</td>
</tr>
<tr>
<td>UCP</td>
<td>Unlike coordination phrase.</td>
</tr>
</tbody>
</table>
Chapter 0

Introduction

言有盡而意無窮。
The words are finite, but their meaning infinite.

Yan Yu (嚴羽; 1191–1241)

Grammar formalisms were developed to answer the question: could language be treated as a mathematical object? In the earliest days of formal syntax, investigators developed formalisms which could delineate the set of grammatical strings in a language with mathematical precision, with decision procedures which could decide grammaticality without recourse to language speakers. However, by formalising the description of language, grammar formalisms also made syntax computable, eventually paving the way for automatic parsing.

Today, computational linguists have at their disposal dozens of grammar formalisms, many of which are interrelated by their theoretical properties, or by common descent.

The focus of this dissertation is to probe the frontier of Combinatory Categorial Grammar (CCG; Steedman, 2000), a lexicalised, mildly context-sensitive grammar formalism, by testing its claims of language-independence (universality) as well as efficiency when applied to the analysis of a new language, Mandarin Chinese.

We achieve this by developing the first abstract analysis of Chinese syntax through the lens of Combinatory Categorial Grammar. We then create Chinese CCGbank, a corpus of ccg derivations reflecting this abstract analysis, which we automatically extract from the Penn Chinese Treebank (Xue et al., 2005). With this new wide-coverage corpus in hand, we conduct the first empirical investigation on Chinese ccg parsing in the literature, by training three state-of-the-art statistical parsers on Chinese CCGbank. Finally, we analyse the errors made by the three parsers, showing that their characteristics on the same dataset
vary considerably. We conclude by pinpointing the sources of error in Chinese parsing which lead to its performance characteristics.

0.1 Why parse?

Parsing, understood as the act of retrieving the latent structure from the linear medium of words, is necessary in many applications which aim to understand natural language as humans do; without this structure, text is merely a sequence of tokens.

Interest in the applications of parsing has seen a resurgence recently, as syntax-directed machine translation techniques have gained prominence as a way to capture complex reordering relationships (Huang et al., 2006). Birch et al. (2007) also incorporated CCG supertags into a factored translation model to improve the quality of reorderings.

The necessity of parsing to deep language understanding also renders it an important component in question answering systems, in which syntactic parsing is widely used to extract dependencies from queries and candidate answers (Shen and Lapata, 2007; Bos et al., 2007). Parsers have also been applied in textual entailment systems to induce logical forms from text (Bos and Markert, 2005; Hickl et al., 2006), and in summarisation systems (McDonald, 2006; Martins and Smith, 2009). These applications all demand accurate, efficient parsing components as prerequisites for syntactic processing.

We focus on Chinese parsing, as we believe a challenge lies in the task which has not been widely investigated in the literature. A gap has long been observed between the states-of-the-art in English (McClosky et al., 2006) and Chinese (Zhang and Clark, 2009). While both figures have climbed over the past decade, a gulf has always separated the two languages. While recent work has brought advances in English parsing to bear on Chinese, little work tries to understand why the gap exists, and what aspects of Chinese syntax are responsible. Being based on a grammar formalism which offers a transparent account of local and non-local dependencies, we intend Chinese CCGbank to be a testbed for this research, a line of investigation begun by our final chapter.

0.2 The thesis

The ultimate goal of this work is to demonstrate the feasibility of efficient Chinese wide-coverage parsing in CCG: the syntactic analysis of general, non-restricted Chinese text. However, to validate the overarching thesis, a number of resources must first be developed.
A pre-requisite to obtain wide-coverage statistical parsing models is an annotated corpus such as a CCGbank; in turn, a pre-requisite for the automatic extraction of a CCGbank is the design of analyses which adequately capture the local and non-local dependency types of a given language. These analyses can be treated as an abstract description of a CCG grammar, independent of the procedure which extracts the corpus. Indeed, due to deficiencies in the source corpus annotation, it may not be feasible to extract the desired analyses automatically without further manual annotation.

**Thesis 1.** An abstract grammar can be developed through CCG to account for the constructions of Chinese syntax, offering a consistent account of local dependencies and the non-local dependencies arising from constructions such as extraction and coordination.

The first contribution of this thesis, therefore, is to show that CCG is capable of accounting for the syntax of Chinese, allowing us to bring a formalism known for its succinct accounts of non-local dependencies to a language in which they are plentiful.

To test whether efficient wide-coverage CCG parsing is possible in Chinese, we first develop a CCG corpus which would serve as the training and evaluation data for a parsing system. To avoid the massive cost of manual annotation, we adopt the corpus conversion methodology, performing an automatic corpus conversion from the Penn Chinese Treebank (PCTB; Palmer et al., 2007) to realise our abstract grammar.

**Thesis 2.** The corpus conversion methodology can be adapted to automatically extract a wide-coverage corpus of Chinese CCG derivations from the Penn Chinese Treebank, reifying an abstract CCG grammar.

Two prominent reasons to choose CCG as the grammar formalism underlying this work relate to non-local dependencies (NLDs) — word-word dependencies which can be separated by an arbitrary distance in a sentence. In particular, 1) CCG has been shown to recover non-local dependencies transparently, offering succinct analyses of many constructions, and 2) Chinese text has been shown to generate NLDs more often than in English (Guo et al., 2007b). However, due to differences in the underlying theory, the conversion of PCTB-style constituent trees to CCG derivations is not straightforward.

The analysis of non-local syntax in PCTB is an instantiation of government and binding theory (Chomsky, 1993), which treats non-local dependencies as arising from movement. On the other hand, CCG, as a *monostratal* or one-layer grammar formalism, analyses non-local dependencies as arising transparently from the combination of overt lexical items.
Thesis 3. The CCG analyses of extraction, passivisation and other non-local phenomena can be extracted automatically from the annotation in the Penn Chinese Treebank, validating its utility as a source corpus for corpus conversion.

We demonstrate, by adapting the CCGbank corpus conversion methodology, that it is possible to automatically extract our chosen abstract CCG analyses from Penn Chinese Treebank trees, fulfilling the requirement for a wide-coverage corpus of Chinese CCG derivations. The second contribution of this thesis is hence Chinese CCGbank, a corpus of 750,000 words automatically converted from the PC Stem through corpus conversion.

Thesis 4. Efficient, accurate wide-coverage Chinese parsing in CCG is possible. The benefits of deep grammar formalisms, which offer accounts of local and non-local dependencies, can be brought to a language where non-local dependencies are particularly prevalent.

Efficient wide-coverage CCG parsing has already been demonstrated for English by Hockenmaier (2003), who first showed that a generative parsing model equipped with the richer dependency representation bestowed by CCG could offer substantial accuracy gains, and Clark and Curran (2007b), who developed C&C, an efficient, wide-coverage CCG parser which demonstrated state-of-the-art results for English dependency retrieval. Both of these parsers were trained on CCGbank (Hockenmaier, 2003), a corpus of CCG derivations automatically extracted from constituent trees in the Wall Street Journal section of the Penn Treebank (Marcus et al., 1994).

To demonstrate the thesis itself, we take Chinese CCGbank, the newly developed corpus, and use it as the training data for three state-of-the-art parsers, which differ by architecture, machine learning technique, and parsing algorithm, showing empirically that the Chinese CCGbank annotation procedure of Chapters 4 and 5, and therefore, the abstract analysis of Chapters 2 and 3, successfully capture the syntax of Chinese. We measure accuracy using a dependency-based metric which is stringent and sensitive to the ability of parsers to recover NLDS, which are more common in Chinese (Guo et al., 2007b).

We uncover surprising differences between the performance of the three Chinese parsers, which were not observed for English. Our final contribution quantifies the contribution of several characteristics of Chinese text to the observed parsing results. We argue that understanding the differences between Chinese and English syntax which render particular NLP techniques more effective will be critical in closing the Chinese-English gap.
0.3 The structure of this work

The thesis is structured as four pairs of chapters. Chapter 0 introduces the questions which this dissertation answers and the contributions of this work. Chapter 1 presents the machinery of CCG and the problems that it solves.

Then, we present our abstract analysis of Chinese syntax through CCG in two chapters. Chapter 2 accounts for its basic word order and modifier-taking behaviour, the choice of atomic syntactic categories, coordination and verb compounding behaviour. Having accounted for the local syntax of Chinese, Chapter 3 gives the first CCG accounts in the literature of special syntactic constructions including the 把/被 ba/bei constructions, extraction, topicalisation and other instances of non-local syntax.

We then describe Chinese CCGbank, the result of a corpus conversion from the PCGB which reifies the abstract analysis, a pair of chapters mirroring the previous two. Chapter 4 describes the characteristics of the source corpus, and presents the framework of the corpus conversion algorithm, allowing the conversion of the local syntax described in Chapter 2. Then, Chapter 5 focuses on extracting the analyses of special syntax developed in Chapter 3 from the PCGB.

The final pair of chapters is an empirical study of Chinese CCGbank through parsing. Chapter 6 validates the thesis of this work, demonstrating accurate, efficient Chinese CCG parsing, by training three architecturally distinct parsers on the newly obtained Chinese CCGbank, producing the first empirical results for statistical CCG parsing on Chinese data.

We discover that while the three parsers achieved similar performance on English data, the models trained on Chinese CCGbank have strikingly different properties. Chapter 7 investigates aspects of Chinese which explain the observed behaviour, introducing the methodology of parser-aided corpus conversion for guiding corpus annotation decisions through parsing, and investigating the role of tagger error and Chinese syntax.

Figure 1: A roadmap of this thesis
0.4 Notation

We follow several typographical distinctions in this work to distinguish various kinds of objects represented by strings:

<table>
<thead>
<tr>
<th>Representation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td>Green italics</td>
</tr>
<tr>
<td></td>
<td>A ccg category</td>
</tr>
<tr>
<td>NP</td>
<td>Blue roman</td>
</tr>
<tr>
<td></td>
<td>A phrase structure label</td>
</tr>
<tr>
<td>NP</td>
<td>Monospace</td>
</tr>
<tr>
<td></td>
<td>A treebank label</td>
</tr>
<tr>
<td>word</td>
<td>Sans serif</td>
</tr>
<tr>
<td></td>
<td>An orthographic string</td>
</tr>
<tr>
<td></td>
<td>of English</td>
</tr>
</tbody>
</table>

Table 1: Orthographic conventions

Glosses and the abbreviations used are based on the Leipzig Glossing Rules (Bickel et al., 2004). A doubled question mark preceding an example indicates questionable grammaticality; an asterisk indicates ungrammaticality.

(1)  a. ??John is a man who dislikes dogs but cats love.
     b. *John is a man who he dislikes dogs but cats love him.

Unless otherwise qualified, all references to “Chinese” refer to the modern Standard Mandarin Chinese language, known as 普通话 putonghua in the People’s Republic of China and elsewhere as 国语 guoyu or 华语 huayu. Language examples are presented in simplified characters, in common with the Penn Chinese Treebank, and for the transcription of Chinese characters, we adhere strictly to Hanyu Pinyin.¹

¹GB/T 16159-1996.
0.5 References to the Penn Chinese Treebank

When we draw an example from the corpus, the section, document and derivation number is reproduced in the margin to facilitate looking up the original derivation. A derivation specifier `xx:yy(d)` corresponds to derivation `d` of file `chtb_xxxy.fid` in the Penn Chinese Treebank.

(2) 外商 投资 企业 成为 中国 外贸 重要 增长点
foreign.business investment project become China foreign.trade important growth.area

Foreign investment projects become an important growth area for China

0.6 Acquiring a copy of Chinese CCGbank

Due to the licensing agreements in place for distributing the Penn Chinese Treebank, we can currently only supply the conversion algorithm, which can be applied to a licensed copy of the Penn Chinese Treebank to obtain a copy of Chinese CCGbank.

If you would like a copy, please visit http://overpunch.com/cnccg.html.

0.7 Publications

Material in Chapters 2–5 was presented in 2010 at the 23rd International Conference on Computational Linguistics (COLING) in Beijing, China (Tse and Curran, 2010), and material in Chapter 6 and 7 was presented at the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) in Montréal, Canada (Tse and Curran, 2012).
Grammar formalisms are formal systems for specifying the syntax of languages. They model the ability of language users to recognise grammatical strings, generate new strings and assign semantic interpretations to strings in a mathematically precise way, conferring these abilities to computational systems. Since grammar formalisms make language computable, they form a central topic in natural language processing.

This chapter introduces Combinatory Categorial Grammar (Steedman, 2000), the lexicalised grammar formalism at the heart of this work. We introduce the machinery of CCG used later to analyse many of the familiar constructions of Chinese syntax, and outline some of the challenges to which CCG provides a solution.

We begin by considering a precursor to Combinatory Categorial Grammar — the pure categorial grammar of Ajdukiewicz (1935) and Bar-Hillel (1953), also known as AB categorial grammar. This allows us to characterise some of the representation issues in pure categorial grammar which led researchers to extend the formalism in diverse ways, one of which was Combinatory Categorial Grammar.
1.1 A gentle introduction to CCG through categorial grammar

We start with a story of two trees. The tree on the left is a derivation of a phrase structure grammar (psg), representing the incremental hierarchical grouping of the words green steak into a noun phrase, then the words eats green steak into a verb phrase, then finally, Bec eats green steak into a full sentence.

The categorial grammar (cg) derivation on the right is a re-labelling of the phrase structure derivation, with some additional projections inserted; the three pairs of equivalent production rules show that the same combinations are achieved in either case.1

But as we will see, the cg labels, or categories, are a lot more descriptive than the opaque non-terminals in the psg production rules: categories declare how they form larger units through the act of combining with the categories of other words.

For instance, the category of an English intransitive verb (a word which seeks a leftward subject NP to form a sentence) succinctly encodes its argument-taking behaviour:

\[ \text{sleep} \vdash S[dcl] \backslash NP \]

We can read this category as follows: sleep is a word which combines with an NP to the left (\( \backslash NP \)) to form a declarative sentence (S[dcl]).2

---

1 Note that an additional unary production \( N \rightarrow NP \) occurs in the cg derivation with no analogue in the psg derivation. The applicability and effect of such type-change rules will be a recurring theme of this thesis. Further discussion appears in §1.8, 1.9, and throughout.

2 The annotation [dcl] says that the sentence is a statement, rather than, for instance, a yes-no question (S[q]). We formally introduce and justify this feature mechanism in §1.5.1.
Then, the category of a transitive verb (a word which additionally seeks a rightward object $NP$) can be built up from the intransitive verb category by adding a rightward $NP$ argument ($/NP$):

$$\text{eat} \vdash (S[\text{dcl}]\backslash NP)/NP$$

And finally, the category of a ditransitive verb (a word which takes two rightward $NPs$ representing the recipient and the theme) builds on the category of the transitive verb, since it takes an additional argument:

$$\text{give} \vdash ((S[\text{dcl}]\backslash NP)/NP)/NP$$

We can see that category structure is recursive, and built from a small number of atoms (like $S[\text{dcl}]$ and $NP$) to form complex categories (like $(S[\text{dcl}]\backslash NP)/NP$).

Now, we can see that the three CCG rules (the bottom row of the production rules on page 12) are actually instantiations of the same two meta-rules, known as the combinatory rules:

$$X/Y Y \rightarrow X \quad (>)$$
$$Y X\backslash Y \rightarrow X \quad (<)$$

To see these at work, let’s consider the rule instantiation $NP \ S[\text{dcl}]\backslash NP \rightarrow S[\text{dcl}]$. The rule ($<$) says that if I have one category $S[\text{dcl}]\backslash NP$, and another category $NP$ to its left, we can derive the category $S[\text{dcl}]$ by combining the two categories using the backward application combinatory rule ($<$). We can depict this combination as follows:

$$\text{(1)} \quad \begin{array}{c} \text{Bec eats green steak} \\ \hline \text{NP} \quad S[\text{dcl}]\backslash NP \end{array} \quad S[\text{dcl}] \quad <$$

Similarly, the rule instantiation $N/N \ N \rightarrow N$ says that if I have the category $N/N$, and another category $N$ to its right, the whole span has category $N$:

$$\text{(2)} \quad \begin{array}{c} \text{green steak} \\ \hline \text{N/N} \quad \text{N} \end{array} \quad \text{N} \quad >$$
Consider that, in a PSG, we must consult the production rules to know that a verb (VV) can combine with a direct object (NP) to form a verb phrase (VP), or that an adjective (JJ) may combine with a noun (NN) to form a noun phrase (NP).\(^3\)

CG, however, avoids the need for a large set of production rules. Instead, the labels themselves (the categories) specify their own argument-taking behaviour, with combinatory rules like (>\(>)\) and (<\(<\)) operating independently of the grammar.

In other words, the lexicon fully specifies the grammar, and so categorial grammars are known as lexicalised grammar formalisms. A full derivation of Bec eats green steak begins with an assignment of categories to the input tokens, followed by the successive application of combinatory rules.

(3)  
\[
\begin{array}{ccc}
\text{Bec} & \text{eats} & \text{green} \\
N & N & N \\
\text{NP} & \text{NP} & \text{NP} \\
\text{S[dl]}\backslash\text{NP} & \text{S[dl]}\backslash\text{NP} & \text{S[dl]}\backslash\text{NP} \\
\end{array}
\]

Aside from pushing the information encoded in production rules into the labels, doing away with production rules altogether, CG still doesn’t seem to give us much over a plain PSG. One of its advantages lies in the interpretation of the combinatory rules.

Our statement of the combinatory rules (>\(>)\) and (<\(<\)) is not quite complete. Each combinatory rule also has a corresponding semantic operation. Just as \(NP \ S[dl]\backslash NP \rightarrow S[dl]\) tells us that the syntactic units NP and \(S[dl]\backslash NP\) can be combined, the rule also tells us how the semantics corresponding to the left and right hand sides can be combined in a principled way. We already know that the word eat has category \(S[dl]\backslash NP)/NP\). We can write out its full lexicon entry, complete with its semantics, which we specify as \(\lambda\)-terms (Church, 1932):\(^4\)

\[
\text{eat} \vdash (S[dl]\backslash NP)/NP : \lambda \text{agent} . \lambda \text{patient} . \text{eat agent patient}
\]

---

\(^3\) In this section, we use the Penn Chinese Treebank POS tags (W for verbs) rather than Penn Treebank POS tags (VBZ, VBD, \ldots) for consistency with our later discussion.

\(^4\) The primed atoms (eat‘) distinguish the orthographic form of the word (eat) from an opaque representation of its meaning (eat‘).
The two rules $(>)$ and $(<)$ are known as forward and backward application, and if we similarly extend the combinatory rules $(>)$, $(<)$ with their semantics:

\[
\begin{align*}
X/Y : f \quad Y : a & \rightarrow X : fa \quad (>) \\
Y : a \quad X/Y : f & \rightarrow X : fa \quad (<)
\end{align*}
\]

we can see why. The semantic interpretation of the application combinatory rules is simply function application of the functor to its argument.

In categorial grammars, the composition of syntax is the composition of semantics, a property sometimes known as its transparent syntax-semantics interface (Steedman, 2000, p.1). Assembling the constituents of Bec eats green steak also assembles its logical form eat’ Bec’ (green’ steak’):

The categorial grammar we have presented, with only the combinatory rules $(>)$ and $(<)$, is weakly equivalent to context-free grammar, in that both formalisms are only capable of specifying the same sets of strings (Bar-Hillel et al., 1960). This application-only categorial grammar is sometimes called AB categorial grammar, after Ajdukiewicz (1935) and Bar-Hillel (1953).

Steedman observed that by increasing the generative power of the AB categorial grammar just a little, we can capture additional syntactic phenomena which context-free grammar struggles to generalise over in a principled manner. He did this by augmenting the system to include combinatory rules whose semantics corresponded to two of the well-studied combinators of combinatory logic: composition (written B) and type-raising (T), resulting in Combinatory Categorial Grammar (CCG). Judicious use of these combinators endows CCG with the ability to capture a plethora of cross-linguistic constructions including extraction, non-constituent coordination, the argument cluster coordination, and
various other movement phenomena. The increase in generative power imbued by these combinators situates CCG between the context-free and context-sensitive languages, and CCG can be identified with a larger group of formalisms known as *mildly context-sensitive grammar formalisms* (Weir, 1988).

Coordination (X and Y; X or Y) is understood to be possible when X and Y have the same syntactic type, explaining the acceptability and unacceptability of:

\[(5) \quad \begin{align*}
  & \text{a. } [\text{a brown paper bag}]_{NP} \text{ and } [\text{a T-bone steak}]_{NP} \\
  & \text{b. } *[\text{a brown paper bag}]_{NP} \text{ and } [\text{exceedingly happy}]_{ADJP}
\end{align*}\]

This constraint on coordination explains why two NPs can be coordinated, while an NP and an adjective phrase (ADJP) cannot. However, these coordinations are also possible:

\[(6) \quad \begin{align*}
  & \text{a. } [\text{[a brown] and [an orange]] paper bag} \\
  & \text{b. } [\text{[I ate] but [you devoured]] the pumpkin.}
\end{align*}\]

despite the fact that a brown or you devoured are not considered full constituents in PSG, since they both have a “hole” where a noun or noun phrase would usually go.

This is a problem for a PSG, which would have to entertain the unappealing notion of hypothesising a production rule for every possible “hole” in a given production rule, in order to predict that coordination is possible when each of the conjuncts have a “hole” in the same place.

The additional combinators B (composition) and T (type-raising) come to the rescue, allowing CCG to assign a principled category (and semantic interpretation) to any partial constituent, including those resulting from non-constituent coordination.

\[(7) \quad \begin{align*}
  \text{I ate but you devoured pumpkins} & \\
  \text{NP} & \begin{array}{c}
    (S[\text{dcl}]\text{NP})/\text{NP} \\
    S/(S\text{\text{NP}}) \rightarrow^{T} \text{NP}
  \end{array} & \begin{array}{c}
    \text{NP} \\
    S/(S\text{\text{NP}}) \rightarrow^{T} \text{NP}
  \end{array} & \begin{array}{c}
    (S[\text{dcl}]\text{NP})/\text{NP} \\
    S[\text{dcl}]/\text{NP} \rightarrow^{B} \text{NP}
  \end{array} & \begin{array}{c}
    \text{NP} \\
    S[\text{dcl}]/\text{NP} \rightarrow^{B} \text{NP}
  \end{array} & \begin{array}{c}
    \text{NP} \\
    S[\text{dcl}]/\text{NP} \rightarrow^{(B)} \text{NP}
  \end{array} & \begin{array}{c}
    \text{NP} \\
    S[\text{dcl}]/\text{NP} \rightarrow^{(B)} \text{NP}
  \end{array} & \begin{array}{c}
    \text{NP} \\
    S[\text{dcl}]/\text{NP} \rightarrow^{(B)} \text{NP}
  \end{array}
\end{align*}\]

In summary, three key advantages of Combinatory Categorial Grammar are that:

---

\(5\) These are named after the composition and type-raising combinators from combinatory logic, but Smul-lyan (1985) knew them as the bluebird (B) and the thrush (T) of his combinatory forest.
1.2. The development of grammar formalisms

- *the lexicon fully specifies the grammar*, so differences in syntax can be expressed on the granularity of words;

- *the composition of syntax is the composition of semantics*, so the construction of syntactic structure and semantic structure occurs in parallel, and

- *mild context-sensitivity* unlocks elegant analyses of cross-linguistic special syntax.

We now proceed to formally define Combinatory Categorial Grammar, its place among modern grammar formalisms, and some of its applications in the NLP literature.

1.2 The development of grammar formalisms

A grammar formalism is a meta-language for the specification of the set of valid language strings of a language. Historically, grammar formalisms have developed along two lines: 1) obtaining the underlying semantics or logical form of a language string; and 2) describing the generation of the string, indirectly constraining the set of possible language strings.

A grammar formalism was first used by the grammarian Pāṇini to characterise the morphology and syntax of Classical Sanskrit, using rewrite rules in a manner rediscovered only much later by Backus (1959), to characterise the syntax of an altogether different kind of language, the programming language ALGOL-58.

The influential methodology of de Saussure’s structural linguistics treated language as a mechanistic system, yet formal methods of description eluded structural linguists. The (re)birth of formalism in grammar began with Post’s canonical systems (1943), Chomsky’s context-free grammars (1957), and the various categorial grammars of Ajdukiewicz (1935), Bar-Hillel (1953) and Lambek (1960). These all characterise language-like systems with a mathematical precision which would ultimately prove useful for automatic natural language parsing.

Combinatory Categorial Grammar (Steedman, 2000), on which we model our analysis of Chinese syntax, is a descendant of the categorial grammars developed by Ajdukiewicz and Bar-Hillel. Categorial grammar is an instance of a *lexicalised* grammar formalism — where ideally, the source of language-dependent variation lies solely or primarily in the lexicon. An instance of a context-free grammar is specified as a series of context-free rewrite rules, the production rules. By contrast, an instance of a categorial grammar consists solely of the mapping of lexical items to *categories*, with a small kernel of language-independent operations belonging instead to the theory. Lexicalisation allows variations in syntax to be expressed at the granularity of individual lexical items.
Chomsky (1957), observing that phenomena such as verbal agreement and passivisation could not easily be accounted for in context-free grammar, augmented the theory with a transformational component which mapped deep structure trees to surface structure trees. Bar-Hillel et al. (1960) proved the equivalence in weak generative power between pure categorial grammar and context-free grammar, leading to diminished interest in the formalism, in line with contemporary linguists’ belief that the analysis of cross-linguistic syntactic phenomena required strictly more than context-free generative power.

In the 1980s, the maturation of computational linguistics led computer scientists to seek theories of grammar which could be formalised and hence manipulated by computational methods. Pollard and Sag (1988) articulate this benefit as follows:

One of the most significant contributions of unification-based linguistics has been the development of a common, mathematically precise formulation within which a wide range of distinct theories (and differing hypotheses about a given phenomenon in the context of a fixed theory) can be explicitly constructed and meaningfully compared.

Ades and Steedman (1982) supplemented the pure categorial grammar with combinators from the combinatory logic of Curry et al. (1958). The weak equivalence proof in Joshi et al. (1990) situates the resulting formalism among the class of mildly context-sensitive grammars defined by Weir (1988), along with tree adjoining grammar (Joshi et al., 1975) and linear indexed grammar (Aho, 1968). The resulting formalism was known as Combinatory Categorial Grammar (ccg; Steedman, 2000). By incorporating those combinators necessary to raise the generative power of the theory enough to handle common elements of cross-linguistic syntax, these additional combinatory rules unlocked simpler non-transformational accounts of diverse phenomena including passivisation, right node raising, and non-constituent coordination. This established that categorial grammars were capable of accounting for the diversity of human language syntax.

The development of the Penn Treebank (Marcus et al., 1994), a syntactically annotated corpus encompassing 1.25 million words of Wall Street Journal text, almost 1.2 million words from the Brown Corpus (Francis and Kucera, 1979), and a number of other domain-specific text types, first enabled researchers to learn wide-coverage parsing models from text. The development of the Penn Treebank involved a massive annotation effort, and proponents of grammar formalisms such as Head-driven Phrase Structure Grammar (HPG;
1.3. The construction of categories

In CCG, the category of a span of tokens is a compact representation of what it may combine with and what it produces after having combined with its arguments: in other words, its functional type.

The lexicon is then the mapping from lexical items to categories, indicating for each lexical item its argument-seeking behaviour. A benefit of lexicalisation was articulated early by Lambek (1958):

\[ \ldots \text{the sentence structure of a formalized language is completely determined by its type list [the lexicon].} \]

The construction of categories begins with the \textit{atoms}. The universe of atoms in classical AB categorial grammar consisted only of $S$ (in Montagovian terms, of type $t$ for \textit{truth value}), and $N$ (of type $e$ for \textit{entity}) (Bar-Hillel, 1953). Given these two atoms, we can construct
Chapter 1. Combinatory Categorial Grammar

Table 1.1: English CCGbank atoms

<table>
<thead>
<tr>
<th>Atom</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Bare noun</td>
<td>(8) friend; ice-cream</td>
</tr>
<tr>
<td>NP</td>
<td>Non-bare noun</td>
<td>(9) John; Bec</td>
</tr>
<tr>
<td>PP</td>
<td>Complement prep. phrase</td>
<td>(10) from the kitchen; inside the box</td>
</tr>
<tr>
<td>S</td>
<td>Sentence</td>
<td>(11) John put the bunny back in the box.</td>
</tr>
<tr>
<td>conj</td>
<td>Coordination word</td>
<td>(12) and; or; ,</td>
</tr>
</tbody>
</table>

grammars by recursively building the atoms into larger categories:

\[
\begin{align*}
  John & \vdash N \\
  never & \vdash (S\backslash N)/(S\backslash N) \\
  works & \vdash S\backslash N \\
\end{align*}
\]

(examples from Lambek (1958))

Yet, a grammar with only these atoms cannot express that particular verbs, such as increase, subcategorise specifically for prepositional phrases. As such, wide-coverage ccg grammars such as English CCGbank typically include PP. The full set of English CCGbank atoms in Hockenmaier (2003) is reproduced in Table 1.1. Later in our analysis of Chinese syntax, we will justify augmentations of the atom inventory along similar lines.

Formally, the set of categories is assembled from a set of atomic categories:

**Definition 1.** (from Steedman (2000)) Given a finite set of atomic categories \( \mathcal{F} \), the set of categories \( \mathcal{C} \) is the smallest set such that:

- \( \mathcal{F} \subseteq \mathcal{C} \)
- \( X/Y, X\backslash Y \in \mathcal{C} \) if \( X, Y \in \mathcal{C} \)

For example, if \( \mathcal{F} = \{ S, \ NP \} \), then examples of elements of \( \mathcal{C} \) include the atoms \( S \) and \( NP \), as well as \( (S\backslash NP)/(S\backslash N) \) and \( (NP\backslash NP)\backslash (NP\backslash NP) \). The categories recursively assembled from the atoms are known as functors or complex categories.

In any complex category \( X/Y \) or \( X\backslash Y \), we call \( Y \) the argument category, and \( X \) the result; modifier categories have the form \( X/X \) of \( X\backslash X \) for any category \( X \), leaving the modified
category unchanged. In English, the categories of word classes such as adjectives and adverbs have modifier shape, because adjectives and adverbs respectively modify nouns and verbs:

\[(\mathbf{13}) \begin{align*}
\text{a. } & \text{ green steak} \\
& \begin{array}{c}
N/N \\
N
\end{array} \\
& \rightarrow \\
& \begin{array}{c}
N
\end{array}
\end{align*}
\]

\[\begin{align*}
\text{b. } & \text{ slept soundly} \\
& \begin{array}{c}
S[\text{dcl}] \backslash NP \\
(S \backslash NP) \backslash (S \backslash NP)
\end{array} \\
& \rightarrow \\
& \begin{array}{c}
S[\text{dcl}] \backslash NP
\end{array}
\end{align*}\]

### 1.4 How we depict derivations

A CCG derivation consists of two steps: an assignment of categories to each input token, followed by successive applications of combinatory rules on adjacent category spans. We can depict a derivation in several ways, to specify the category assignments made to the input tokens, or to illustrate the sequence of combinatory rules used to obtain a given analysis.

In a tabular derivation, the input tokens appear in the first row. The next row depicts the category assigned to each token. In each successive row, the combinatory rule indicated by the arrowhead on each horizontal line segment indicates that the given rule has been applied to the span. Combinatory rules continue to apply until we have derived a top-level category which spans the entire input. The tabular derivation can be interpreted as an upside down derivation tree, with the leaves at the top.

Thus far, we have primarily depicted CCG derivations in tabular form, as they make explicit all category assignments and all combinatory rule applications.

A CCG derivation may be represented as a binary-branching tree, as in (1), in which each internal node is the result of a combinatory rule applied to its children. This representation is useful to illustrate the connection between the structure of the source Penn Chinese Treebank trees and the CCG derivations we obtain. Although we omit the particular combinatory rule which applies at each internal node, this should be clear from context.

Finally, we occasionally annotate only those constituents of interest with CCG categories:

---

7 Additionally, the heads within the modifier category’s argument and result must all be coindexed. As such, \((S \backslash NP)_{y} / (S \backslash NP)_{y}\) is a modifier category because the argument and result are structurally identical and share the coindexation \((S\backslash NP)_{y}\); the category of a subject control verb such as \((S[\text{dcl}] \backslash NP)_{y} / (S[\text{dcl}] \backslash NP)_{w}\) is not because the two innermost categories have different indices \(y\) and \(w\). We describe the coindexation mechanism in detail in Section 1.5.4.
(14) \[ \text{Bec}_{NP} \text{ eats green steak }]_{S[dcl]\backslash NP}^{} \]

1.5 Decorations on categories

1.5.1 Features

Lambek (1958) recognised early that while a grammar which does not distinguish grammatical number will over-generate:

(15) a. \[ \text{Men}_N \text{ walk }]_{S\backslash N}^{} \]
b. \[ * \text{Men}_N \text{ walks }]_{S\backslash N}^{} \]

assigning plural nouns a different category, say $N^*$, would lead to a multiplicity of rules reflecting false distinctions, for instance:

(16) \[ \text{red} \vdash N/N, N^*/N^* \]

Unification grammars (Shieber, 1986) offer a solution in which distinctions may be specified when available, yet underspecified when not pertinent. Unification-based formalisms operate over objects called feature structures or attribute-value matrices, introduced in the following section.\(^8\) They also formalise the notion of coindexation or identity of value between different parts of a structure.

Many formalisms, including CCG and HPSG, can be cast as unification-based formalisms, imbuing these representational benefits. Hockenmaier (2003) imports this interpretation, allowing morphological distinctions to be expressed as annotations known as features.

Conventionally, features are depicted as values between square brackets following a category. For instance, the bare category $S$ decorated with the feature $dcl$, which represents declarative mood, is represented as $S[dcl]$.

With features, morphological variants can share a structural category\(^9\), while marking these distinctions in a manner orthogonal to atom identity:

\(^8\) The meaning of “feature” differs in the CCG and HPSG literatures. To avoid confusion in this work, we only refer to structures such as the one in Figure 1.1 as attribute-value matrices rather than feature structures,-reserving “feature” for the CCG category annotation discussed in this section.

\(^9\) Two categories are said to share a structural category if they are the same modulo feature values and other decorations. For instance, $(S[dcl]\backslash NP)/NP$ and $(S[b]\backslash NP)/NP$ share a structural category because they are unifiable with $(S[X]\backslash NP)/NP$. 
1.5. Decorations on categories

(17) a. \textit{dcl} : declarative mood  
b. \textit{b} : bare stem (infinitive) verb form  
c. \textit{pt} : past participle verb form  
d. \textit{ng} : -ing verb form

(18) a. break $\vdash (S[dcl]\backslash NP)/NP, (S[b]\backslash NP)/NP$  
b. broken $\vdash (S[pt]\backslash NP)/NP$  
c. breaking $\vdash (S[ng]\backslash NP)/NP$

Furthermore, modifiers such as \textit{N/N} can underspecify for a feature, avoiding the multiplicity of rules which would otherwise result, while allowing grammar writers to specify fine-grained distinctions in type when necessary (for instance \textit{water} $\vdash N[+\text{mass}, -\text{animate}]$) but to generalise over these distinctions when not necessary (\textit{green} $\vdash N/N$).

An analysis with features allows words to select for particular morphological forms, such as the auxiliary verb \textit{has} in (19), which accepts a verb in past participle form, while avoiding the proliferation of modifier categories which would result if different morphological forms resulted in distinct atoms:

(19) \begin{align*}
\frac{(S[dcl]\backslash NP)/(S[pt]\backslash NP)}{(S\backslash NP)/(S\backslash NP)} & \rightarrow & \frac{(S[pt]\backslash NP)}{(S\backslash NP)} & \rightarrow & \frac{(S[dcl]\backslash NP)}{(S\backslash NP)}
\end{align*}

A final use for the feature mechanism is to control over-generation, by marking special syntax with features, such that the special syntax is blocked outside of a particular syntactic frame. Examples of derivational control through features can be found in the analysis of null complementisers in English subordinate clauses in Steedman (1996), Dutch subordinate clause word order (Steedman, 2000, §6.1), and our own categorial analysis of a Chinese \textit{wh}-word asymmetry to be presented in Chapter 3.

The feature mechanism is subsumed by the much more general notion of attribute-value matrices, which allows us to formally model 1) the augmentation of categories with arbitrary structure, 2) the unification of categories, and 3) the notion of coindexation between sub-parts of a category.
### Attribute Interpretation

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEAD</td>
<td>Head variable or value</td>
</tr>
<tr>
<td>FEAT</td>
<td>Feature AVM</td>
</tr>
</tbody>
</table>

### Attributes of complex categories

| RES      | AVM of result category (X in X|Y) |
| ARG      | AVM of argument category (Y in X|Y) |
| SLASH    | Directionality of category’s slash |

### Attributes of atomic categories

| CAT      | Category atom |

Table 1.2: An AVM model of category structure

### 1.5.2 Categories as attribute-value matrices

As Shieber (1986) notes, attribute-value matrices (AVMs) can be used to model CCG categories; Table 1.2 is one possible model.

This characterisation allows us to formalise the notion of decorating categories with features and head variables, as well as the procedure of unification over categories. All of the decorations on categories discussed in this section can be modelled as additional attributes in a category structure. For instance, the attribute-value matrix in Figure 1.1 corresponds to the category assignment of de ⊢ (NP_y/NP_y) \((S[dcl]\backslash NP_y)\) of the Chinese relativiser.

When two categories combine by a combinatory rule, their common sub-categories are unified according to that rule’s schema. For instance, consider this partial derivation:

\[
\begin{array}{c}
\text{NP}_{15G} \\
\text{S[X]/(S[X]\backslash NP_{15G})} \\
\end{array}
\]

\[
\begin{array}{c}
\text{NP} \\
\end{array}
\]

The type-raised category \(S[X]/(S[X]\backslash NP_{15G})\) is about to combine with the lexical category \((S[dcl]\backslash NP_p)/NP_p\) on the verb 买 buy, through forward application \((X/Y \ Y \rightarrow \text{NP})\).
### 1.5. Decorations on categories

#### 1.5.2. Attribute-value matrix representation

<table>
<thead>
<tr>
<th>RES</th>
<th>CAT</th>
<th>SLASH</th>
<th>ARG</th>
<th>HEAD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RES</th>
<th>CAT</th>
<th>SLASH</th>
<th>ARG</th>
<th>HEAD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1.1: Attribute-value matrix representation of \( \mathcal{d} \mathcal{e} \vdash (NP_\mathcal{y}/NP_\mathcal{y}) \setminus (S[dcl]/NP_\mathcal{y}) \)

Thus, the corresponding sub-categories which will undergo recursive unification are \( S[X] \setminus NP_\mathcal{y} \) and \( S[dcl] \setminus NP_\mathcal{y} \). Unification then yields the unifier \( X = dcl, a = \mathcal{y}, 1SG \).

The unification operation proceeds by recursively comparing the corresponding fields of the attribute-value matrices of its operands. As the base case, if both operands are variables, unification enters them into the same equivalence class of variables. If one operand is a variable (e.g. \( \square \)) and the other is a value, the value is assigned to the variable. If both operands are values, unification fails.

#### 1.5.3 AVM encoding of features

Features may be represented as additional attribute-value pairs in the category’s attribute-value matrix. Categories without a feature value are underspecified for their feature attribute. As a special case, in modifier categories such as the verbal pre-modifier category \( (S\setminus NP)/(S\setminus NP) \), all of the feature variables on the argument and result categories are un-
derstood to be coindexed. For instance, in the feature structure in Figure 1.2, the feature $[\text{MOOD} \ [X]]$ occurs both in the result and argument categories with identical coindexation, such that when the argument category takes on a value for the MOOD feature through uni-
ification, the result category takes on the same value as well.

Figure 1.2: The innermost categories of the argument and result categories have a coindexed feature variable

This allows modifier categories to “pass through” the features on the complex categories that they modify.
For example, the category of the aspect particle 了  in (21) is (S\NP) \ (S\NP), which by the modifier category convention is understood to mean (S[X]\NP) \ (S[X]\NP). When its argument category S[X]\NP is unified with the category S[dl]\NP through backward application, the feature variable coindexation causes the result category to inherit the feature [dl] as well.

### 1.5.4 AVM encoding of head variables

The attribute-value matrix corresponding to atomic and complex categories has a field head which holds the lexical item which is the head of the span of that category. In the marked-up notation for categories, two sub-categories with the same value of head share a variable index:¹⁰

\[
[(NP_y/NP_y)_x \backslash (S[dl]_z/NP_y)_z]_x
\]

Unification of head variables is subsumed by the more general unification operation over feature structures (discussed in Section 1.5.2), in which the identity of the head is simply an attribute. A head variable which holds a lexical item is said to be filled.

To clarify the passing of heads during a derivation, we may occasionally annotate the categories explicitly with variable names and values (with the notation \( z_1 \equiv y_2 \)) indicating that the two head variables \( z_1 \) and \( y_2 \) are being unified:

\[
\begin{array}{c}
\text{the} \\
\left[ NP[nb]_{y_1/N_y_1} \right]_{x_1=\text{the}} \\
\text{meteor} \\
\left[ (N_{y_2} \backslash N_{y_2})_{x_3=\text{that}} \backslash (S[dl]_{z_2} \backslash NP_{y_2})_{z_2} \right]_{x_3=\text{that}} \\
\text{that} \\
[S[dl]_{z_3=\text{fell}} \backslash NP_{y_3}]_{z_3=\text{fell}} \\
\text{fell} \\
\left[ N_{y_2} \right]_{z_3=\text{that}} \\
(y_2 \equiv y_3) \\
\left[ x_2 \equiv y_2 \right]
\end{array}
\]
Certain categories in the lexicon mediate non-local, long distance or unbounded dependencies, so called because the distance between the head and the argument can grow unboundedly large:

\[(\text{23})\]

a. the book, that I wrote \(t_i\)

b. the book, that John believes I wrote \(t_i\)

c. the book, that John believes Mary said I wrote \(t_i\)

d. the book, that John believes Mary said Tom keeps claiming that I wrote \(t_i\)

Such constructions, which rely on this head-passing behaviour of categories, are known to be more prevalent in Chinese than in English (Guo et al., 2007b). One approach used by grammar formalisms to capture non-local dependencies is to allow incomplete constituents to be propagated through an unbounded number of intermediate stages in the derivation. This is achieved through the \(\text{SLASH}\) mechanism in \(\text{HPSG}\) (Pollard and Sag, 1994), and in \(\text{CCG}\) through type-raising (\(T\)) and composition (\(B\)).

Finally, two categories are said to share a structural category if they are the same modulo head variables, feature values and modes. For example, \((S[dc]_z \setminus \text{NP}_y)_z/(S[dc]_w \setminus \text{NP}_y)_w\) and \((S[dc]_z \setminus \text{NP}_y)_z/(S[dc]_w \setminus \text{NP}_y)_w\) share a structural category, because they only differ by head coindexation.

### 1.6 Combinatory rules

While categories declare their argument-seeking behaviour, it is the combinatory rules which act on the categories to form new categories.

The combinatory rules assign principled types to spans of more than one token, and constrain the kinds of combinations which are possible given the input tokens’ categories. Thanks to the close relationship between combinatory rules and the combinators of Curry et al. (1958), each of the combinatory rules has a semantic interpretation which, during the parsing process, allows the act of syntactic derivation to simultaneously build predicate-argument structure.

#### 1.6.1 Application \((\succ,\prec)\)

\[
\begin{align*}
X/Y & \quad Y \quad \rightarrow \quad X \quad (\succ) \\
Y & \quad X/Y \quad \rightarrow \quad X \quad (\prec)
\end{align*}
\]
We have already seen the effect of functional application, which combines the categories of adjacent spans according to their canonical argument-seeking behaviour:

\[(\text{Bec} \quad \text{is chewing} \quad \text{a} \quad \text{green} \quad \text{steak})\]

A categorial grammar equipped only with the two application combinatory rules is known as an AB grammar, after Ajdukiewicz (1935) and Bar-Hillel (1953). As suggested in the introduction, such a grammar is equivalent to a phrase structure grammar where the production rules have been pushed into the lexicon. The equivalence between AB grammar and context-free grammar was proven by Bar-Hillel et al. (1960), formalising the limitations of the application-only categorial grammar.

Consider the partial constituent John bought, which has an NP-shaped “hole” in direct object position:

\[(\text{John} \quad \text{bought})\]

Neither of the application combinatory rules can form the expected category \(S[\text{dcl}]/NP\), because the category of the transitive verb bought declares that it takes its object argument first (/NP) before its subject argument (\NP). Lambek (1958) proposed a combinatory calculus in which categories could arbitrarily be rebracketed — for instance, \((S[\text{dcl}]/NP)/NP \rightarrow (S[\text{dcl}]/NP)/NP\) — which would require the accompanying semantics to be rewritten to match. CCG offers a more principled solution.

### 1.6.2 The rebracketing puzzle

In (25), the category NP of John, being an atomic category, is inert. It cannot consume categories; it can only be consumed. The CCG solution to the rebracketing puzzle begins with a rule which allows inert categories to consume categories.

A verb phrase \((S[\text{dcl}]/NP)\) is a span which forms a sentence \((S[\text{dcl}])\) when combined with a subject NP. But under an alternative analysis, a subject NP can also be thought
of as a span which forms a sentence when combined with a verb phrase. The category assignment to the subject \(NP\) which achieves this alternative analysis is \(S/(S\setminus NP)\).

\[
\frac{\text{John} \quad \text{rode a pony}}{S/(S\setminus NP) \quad S[\text{dcl}]\setminus NP} \quad S[\text{dcl}]
\]

This is not a particular property of John; rather, the transformation \(NP \rightarrow S/(S\setminus NP)\) is possible for any \(NP\), and indeed any choice of category in place of \(S\). This is because the transformation is an instance of a type-raising combinatory rule:

\[
X : a \quad \rightarrow \quad T/(T\setminus X) : \lambda f. f a \quad (>T)
\]

With the rule \((>T)\), John in (25) is no longer inert, although no combinatory rule yet introduced can produce the desired output category:

\[
\frac{\text{John} \quad \text{bought}}{NP \quad (S[\text{dcl}]\setminus NP)/NP} \quad S/(S\setminus NP) \quad S[\text{dcl}]/NP
\]

The last piece of the rebracketing puzzle is the combinatory rule analogue of the functional composition of Curry et al. (1958):

\[
X/Y : f \quad Y/Z : g \quad \rightarrow \quad X/Z : \lambda x. f(gx) \quad (>B)
\]

The semantic interpretations accompanying each rule also assign the expected semantics to the top-level category:

\[
\frac{\text{John} \quad \text{bought}}{NP \quad \lambda p. \lambda a. \text{buy } a p} \quad \frac{(S[\text{dcl}]\setminus NP)/NP \quad \lambda f. f\text{john'}}{S/(S\setminus NP)} \quad \frac{\lambda y. \text{buy } \text{john'}}{S[\text{dcl}]/NP} \quad (>B)
\]

The full complement of type-raising and composition rules, of which \((> B, > T)\) are only two, are what unlocks semantically principled analyses for a range of rebracketing and movement phenomena in CCG.
1.6. Combinatory rules

1.6.3 Composition (B)

\[ \begin{align*}
X/Y & \quad Y/Z \quad \rightarrow \quad X/Z \quad (>B) \\
Y\backslash Z & \quad X/Y \quad \rightarrow \quad X\backslash Z \quad (<B) \\
X/Y & \quad Y\backslash Z \quad \rightarrow \quad X\backslash Z \quad (>B_\times) \\
Y/Z & \quad X\backslash Y \quad \rightarrow \quad X/Z \quad (<B_\times)
\end{align*} \]

The four combinatory rules of composition are divided into the harmonic (>B < B) and crossed (>B_\times < B_\times) composition rules.

The two harmonic composition rules form a rebracketing calculus in the system of combinatory rules, which we demonstrate with the two harmonic composition examples in Figure 1.3. Given the sequence of categories \( C \ B \backslash \ C \ A \backslash B \), Figure 1.3b shows that the top-level category \( A \) can be derived in two ways, reflecting the application-only bracketing \( (C \ B \backslash C) \ A \backslash B \), and the alternative bracketing \( C \ (B \backslash C \ A \backslash B) \) achieved through backward harmonic composition. Figure 1.3d shows that an analogous situation holds for forward harmonic composition. Harmonic composition is order-preserving because, despite the rebracketing, no other rearrangement of the input categories can produce the top-level category \( A \).

Crossed composition is used to analyse heavy NP shift in English, in which the recipient PP swaps positions with the theme NP when the theme NP is sufficiently “heavy” or long, despite the canonical V + PP + NP order (Dave sent a bunny to John) indicated by the category of sent:

\[ \begin{array}{c c c c c}
\text{Dave} & \text{sent} & \text{to} & \text{John} & \text{a bunny in a red package which was quite hard to open} \\
VP & (VP/PP)/NP & PP & NP & \hline
\end{array} \]

In general, the two crossed composition rules allow a functor \( A \vert B \) to come between another functor \( B \vert C \) and its argument \( C \), as pictured in Figures 1.4b and 1.4d. Such constructions were licensed explicitly by a rule known as \textsc{wra}p in an early formulation of categorial grammar (Flynn, 1983). A grammar equipped with crossed composition allows all four
Chapter 1. Combinatory Categorial Grammar

\begin{align*}
\begin{array}{c}
C \rightarrow B \\
B \rightarrow A
\end{array} & \begin{array}{c}
C \rightarrow B \\
C \rightarrow A
\end{array} \\
\begin{array}{c}
A \rightarrow C \\
B \rightarrow A
\end{array}
\end{align*}

(a) Canonical derivation using backward application (\(\prec\))

\begin{align*}
\begin{array}{c}
A \rightarrow B \\
B \rightarrow C
\end{array} & \begin{array}{c}
B \rightarrow C \\
C \rightarrow B
\end{array} \\
\begin{array}{c}
C \rightarrow A \\
A \rightarrow C
\end{array}
\end{align*}

(b) Alternative bracketing using backward harmonic composition (\(\prec \rightarrow \mathcal{B}\))

\begin{align*}
\begin{array}{c}
A \rightarrow B \\
B \rightarrow C
\end{array} & \begin{array}{c}
B \rightarrow C \\
C \rightarrow B
\end{array} \\
\begin{array}{c}
C \rightarrow A \\
A \rightarrow C
\end{array}
\end{align*}

(c) Canonical derivation using forward application (\(\rightarrow\))

\begin{align*}
\begin{array}{c}
B \rightarrow C \\
C \rightarrow A
\end{array} & \begin{array}{c}
B \rightarrow C \\
C \rightarrow A
\end{array} \\
\begin{array}{c}
C \rightarrow A \\
A \rightarrow C
\end{array}
\end{align*}

(d) Alternative bracketing using forward harmonic composition (\(\rightarrow \rightarrow \mathcal{B}\))

Figure 1.3: Harmonic composition licenses rebracketing

\begin{align*}
\begin{array}{c}
B \rightarrow C \\
C \rightarrow A
\end{array} & \begin{array}{c}
B \rightarrow C \\
C \rightarrow A
\end{array} \\
\begin{array}{c}
C \rightarrow A \\
A \rightarrow C
\end{array}
\end{align*}

(a) Canonical derivation using application

\begin{align*}
\begin{array}{c}
B \rightarrow A \\
A \rightarrow C
\end{array} & \begin{array}{c}
B \rightarrow A \\
A \rightarrow C
\end{array} \\
\begin{array}{c}
C \rightarrow A \\
A \rightarrow C
\end{array}
\end{align*}

(b) Reordering of constituents using backward crossed composition (\(\prec \rightarrow \mathcal{B}\))

\begin{align*}
\begin{array}{c}
A \rightarrow B \\
B \rightarrow C
\end{array} & \begin{array}{c}
B \rightarrow C \\
C \rightarrow B
\end{array} \\
\begin{array}{c}
C \rightarrow A \\
A \rightarrow C
\end{array}
\end{align*}

(c) Canonical derivation using application

\begin{align*}
\begin{array}{c}
A \rightarrow B \\
B \rightarrow C
\end{array} & \begin{array}{c}
B \rightarrow C \\
C \rightarrow B
\end{array} \\
\begin{array}{c}
C \rightarrow A \\
A \rightarrow C
\end{array}
\end{align*}

(d) Reordering of constituents using forward crossed composition (\(\rightarrow \rightarrow \mathcal{B}\))

Figure 1.4: Crossed composition licenses reordering

category sequences in Figure 1.4 to derive the same top-level category \(A\), making crossed composition an order-permuting operation.

Steedman (2007) shows by recognising the language \(\{a^n b^n c^n d^n | n > 0\}\) that it is the crossed composition rules which elevate the generative power of CCG beyond context-free to mildly context-sensitive.
1.6. Combinatory rules

1.6.4 Type-raising (T)

\[
\begin{align*}
X \rightarrow & \quad T/(T\backslash X) \quad (>\ T) \\
X \rightarrow & \quad T\backslash(T/X) \quad (<\ T)
\end{align*}
\]

In the application-only setting, only functors may consume their arguments. The type-raising rules allow arguments to consume their functors instead.

It was quickly observed that while type-raising is necessary to capture non-canonical argument-taking orders, and hence necessary to generate so-called “non-constituents” in constructions such as coordination and relativisation, the derivation of A from \(A/B \ B\) or \(B \ A\backslash B\) can proceed in two ways. As in Figure 1.5a, the category \(A/B\) may consume \(B\) directly through forward application, or, as in Figure 1.5b, the category \(B\) is type-raised to \(A\backslash(A/B)\) and consumes the category \(A/B\) to its left. Both derivations derive the same top-level category \(A\).

(a) Canonical derivation using forward application (>)

(b) Alternative derivation using backward type-raising (< T)

(c) Canonical derivation using backward application (<)

(d) Alternative derivation using forward type-raising (> T)

Figure 1.5: Type-raising leads to multiple “spurious” derivations

Such ambiguity was termed “spurious ambiguity” because a parser must consider many alternatives which are derivationally distinct but generate the same logical form (Wittenburg, 1987). This observation led to concerns that parsing in cCG was infeasible, until Eisner (1996) demonstrated constraints which, placed on the input categories to the composition rules, can ensure that a parser generates only one member of each equivalence class, eliminating the spurious ambiguity caused by type-raising and composition.\(^{11}\)

\(^{11}\) However, Hockenmaier and Bisk (2010) point out that Eisner normal form fails to preserve all interpretations when the degree of composition is bounded, and also present a normal form with this property.
The normal form consists of two constraints on the products of the composition (B) combinatory rules.

**Normal form constraint 1.** No category output by a rightward composition rule may serve as the left input category to a rightward composition or application rule.

**Normal form constraint 2.** No category output by a leftward composition rule may serve as the right input category to a leftward composition or application rule.

In (30), an analysis in which forward harmonic composition (>B) is applied to Bec likes steak is blocked, because it would require steak to be consumed by forward application.

\[
\begin{array}{c|c|c|c}
\text{Bec} & \text{likes} & \text{steak} \\
\hline
\text{NP} & (S[\text{dcl}] \backslash \text{NP})/\text{NP} & \text{NP} \\
\hline
S/(S\backslash \text{NP}) & & >B \\
\hline
\end{array}
\]

However, in the so-called *non-constituent coordination* (Milward, 1994), in which spans that do not form traditional phrase structure constituents, such as NP + V or Det + Adj, can be coordinated, the normal form constraints are not violated, since the results of composition I borrowed and you bought are never directly inputs to forward application or composition.

\[
\begin{array}{c|c|c|c|c|c|c|c|c|c}
\text{I} & \text{borrowed} & \text{and} & \text{you} & \text{bought} & \text{the} & \text{textbook} \\
\hline
\text{NP} & (S[\text{dcl}] \backslash \text{NP})/\text{NP} & \text{conj} & (S[\text{dcl}] \backslash \text{NP})/\text{NP} & \text{NP}[\text{nb}] / \text{N} & \text{N} \\
\hline
S/(S\backslash \text{NP}) & & & S/(S\backslash \text{NP}) & & & & \\
\hline
\text{S[\text{dcl}]/NP} & >B & \text{S[\text{dcl}]/NP} & >B & \text{S[\text{dcl}]/NP} & \text{S[\text{dcl}]/NP} & > \\
\hline
\end{array}
\]

### 1.6.5 Substitution (S)

\[
\begin{align*}
(X/Y)/Z & \quad Y/Z & \rightarrow & \quad X/Z \quad (>S) \\
Y/Z & \quad (X\backslash Y)/Z & \rightarrow & \quad X\backslash Z \quad (<S) \\
(X/Y)/Z & \quad Y/Z & \rightarrow & \quad X\backslash Z \quad (>S_x) \\
Y/Z & \quad (X\backslash Y)/Z & \rightarrow & \quad X/Z \quad (<S_x)
\end{align*}
\]
These combinatory rule analogues of the substitution combinator \( S \) were introduced by Steedman (1987) and Szabolcsi (1989) to analyse parasitic gap constructions, in which a parasitic gap (\( pg \)) can appear in positions which usually constitute islands for extraction (Culicover and Postal, 2001).

(32) This is a kind of cake, that you can eat \( t_i \) without baking \( p_{g_i} \).

The purpose of the substitution rules is to allow one category to serve as an argument to two other categories, the desired behaviour in the analysis of parasitic gaps, in which one constituent (e.g. cake) fills two slots (the object of eat as well as baking).

(33) \[
\begin{array}{c}
\text{cake that you can eat without baking} \\
\text{NP} \quad (\text{NP}\backslash\text{NP}) \quad (\text{S[dl]}\backslash\text{NP}) \\
\text{NP} \quad (\text{S[dl]}\backslash\text{NP}) \quad (\text{(S}\backslash\text{NP})(\text{S}\backslash\text{NP})\backslash\text{NP})
\end{array}
\]

\[
\frac{S/(S\backslash NP)}{S/(S\backslash NP)} \quad (\text{S[dl]}\backslash\text{NP}) \quad (\text{(S}\backslash\text{NP})(\text{S}\backslash\text{NP})\backslash\text{NP})
\]

\[
(\text{NP}\backslash\text{NP}) \quad (\text{NP}\backslash\text{NP}) \quad (\text{NP}\backslash\text{NP})
\]

\[
\frac{NP}{NP}
\]

1.6.6 Stipulations on combinatory rules

Steedman (2000) hypothesises a number of universals which constrain the space of possible combinatory rules. Intuitively, these stipulations on combinatory rules state that the rules must preserve the directionality indicated by slashes in the lexicon, and that the rules must apply only to finitely many adjacent elements.

**Principle 1. The principle of adjacency.** Combinatory rules may only apply to finitely many phonologically realised and string-adjacent entities.

The Principle of Adjacency precludes rules which operate on empty categories, or any element which is not phonologically realised, setting CCG apart from formalisms derived from transformational grammar which rely heavily on traces and movement to account for non-local dependencies.

**Principle 2. The principle of consistency.** All syntactic combinatory rules must be consistent with the directionality of the principal function.

**Principle 3. The principle of inheritance.** If the category that results from the application of a combinatory rule is a functor, then the slash defining directionality for a given
argument in that category will be the same as the one(s) defining directionality for the corresponding argument(s) in the input functor(s).

The principles of consistency and inheritance preclude rules which ignore the directionality specified in the lexicon, for instance this non-order-preserving variant of application:

\[ X \downarrow Y Y \rightarrow X \]

whose presence would allow leftward arguments to be consumed on the right hand side, obliterating directionality distinctions in the lexicon:

\[
\begin{array}{c}
\text{slept} \quad \text{John} \\
\hline
S[\text{dcl}] \backslash \text{NP} \\
\text{NP} \quad \downarrow \\
S[\text{dcl}] \\
\end{array}
\]

1.6.7 Stipulations on the lexicon

**Principle 4. The principle of head categorial uniqueness.** A single non-disjunctive\(^{12}\) lexical category for the head of a given construction specifies both the bounded dependencies that arise when its complements are in canonical position and the unbounded dependencies that arise when those complements are displaced under relativization, coordination and the like.

This principle expresses the desire for a given word form to have a single category for each embedding, with the CCG combinators handling the special analysis when that word’s arguments are displaced from their usual positions under syntactic phenomena such as extraction or coordination. However, the principle does *not* stipulate that each lexical item should have one category, or even that each sense should have one category. Rather, it states that in a given embedding (for instance, main clause order versus subordinate clause order in German), syntactic phenomena which cause the head’s arguments to be displaced should be handled not through lexical ambiguity, but through the combinatory rules.

**Principle 5. The principle of lexical head government.** Both bounded and unbounded syntactic dependencies are specified by the lexical syntactic type of their head.

The principle of lexical head government states that CCG is lexicalised and head-driven. The category of the head specifies the dependencies which hold between its arguments.

\(^{12}\) This definition is quoted from Steedman (2000). We take *non-disjunctive* to mean that the condition applies to the mapping of a single lexical item to a single category; it is still possible for a single lexical item to be mapped to multiple categories which yield different dependencies.
1.7 Multi-modal CCG

The fact that enabling the full complement of combinatory rules to operate unconstrained in a language leads to over-generation phenomena such as scrambling was noted early by Steedman (1996). For instance, the forward crossed composition rule (>Bₜ), if enabled in English, results in undesirable word orders:

\[
\begin{array}{cccc}
NP & S/S & (S[\text{dcl}]\backslash NP)/NP & NP \\
& & S[\text{dcl}]\backslash NP & \rightarrow Bₜ \\
& & S[\text{dcl}]\backslash NP & <
\end{array}
\]

In Steedman (2000), combinatory rules which result in over-generation in a given grammar are deemed to be inactive in that grammar. However, base CCG offers no mechanism to specify in the lexicon which rules are active or inactive in that grammar, detracting from its goal, as a lexicalised grammar formalism, to allow all language-specific variation to be specified in the lexicon.

In addition, Baldridge (2002) demonstrates cases where a blanket restriction on a rule will under-generate, while always permitting the rule to operate will over-generate:

\[
\begin{array}{cccc}
\text{a.} & \text{[ strong coffee ]}_N \text{[ with milk ]}_N \\
\text{b.} & \text{[ strong with milk coffee ]}_N \\
& N/N & (N\backslash N)/N & N \\
& N\backslash N & N\backslash (N/N) & N \\
& N\backslash (N/N) & N & N \\
& N & N & <B \\
& N & N & <T \\
& N & N & <B \\
& N & N & <T
\end{array}
\]

The solution of Baldridge (2002) is a mechanism called multi-modal CCG, in which each slash of each category licenses the combinatory rules with which it may be used. This solves the two aforementioned issues: rule restrictions are lexically specified, and can be applied at the granularity of slashes, allowing the lexical items involved in particular constructions to selectively enable or disable rules.

\[\text{13 Backward harmonic composition (< B) is required in English for the CCG analysis of argument cluster coordination (Steedman, 2000), so the rule cannot be de-licensed in the English grammar without de-licensing argument cluster coordination as well.}\]
Baldridge’s formulation of MMCCG proposes a system of seven modes, which are represented as annotations on the slashes of a category. In this work, and the description which follows, we consider a simpler system of four modes, which collapses a distinction made by Baldridge (2002) for capturing the direction-sensitivity of an extraction asymmetry.

Figure 1.6: Simplified mode hierarchy

Baldridge and Kruijff (2003) arranges the modes in a hierarchy (Figure 1.6), such that an arc connecting mode $s$ to mode $t$ denotes that the set of combinatory rules licensed for mode $s$ is a subset of the rules licensed for mode $t$. The topmost mode ($\ast$) is the most restrictive, only allowing application:

$$X/\ast Y \quad Y \Rightarrow X$$  \quad (>\ B)  \quad (1.1)$$

$$Y \quad X\!\setminus\!\ast Y \Rightarrow X$$  \quad (<\ B)  \quad (1.2)$$

The application-only mode simulates the $\text{ab}$ combinators, disabling context-sensitivity altogether when its presence would over-generate. These rules stipulate that for a category to serve as input to application, it must either carry the mode $\ast$, or by the inheritance condition, be any descendant of $\ast$ in the directed graph.

The second tier contains a mode $\odot$ which permits harmonic composition, together with the application rules inherited from mode $\ast$. Similarly, the mode $\times$ permits crossed composition and the application rules:

$$X/\odot Y \quad Y/\odot Z \Rightarrow X/\odot Z$$  \quad (>\ B,\text{c})  \quad (1.3)$$

$$Y\!\setminus\!\odot Z \quad X\!\setminus\!\odot Y \Rightarrow X\!\setminus\!\odot Z$$  \quad (<\ B,\text{c})  \quad (1.4)$$

$$X/\times Y \quad Y\!\setminus\!\times Z \Rightarrow X\!\setminus\!\times Z$$  \quad (>\ B,\text{c})  \quad (1.5)$$

$$Y/\times Z \quad X\!\setminus\!\times Y \Rightarrow X/\times Z$$  \quad (<\ B,\text{c})  \quad (1.6)$$

Enabling only harmonic composition enables the order-preserving re-bracketing facility of CCG, but critically rules out the order-permuting crossed composition rules.
The bottom-most tier contains the maximally permissive mode \( \cdot \) which is equivalent to unmarked slashes in the base CCG formalism.

Finally, the type-raising combinator, which introduces two slashes, is slightly modified in multi-modal CCG:

\[
X \Rightarrow T/_{i}(T\backslash_{i}X) \quad (> T) \quad (1.7)
\]
\[
X \Rightarrow T\backslash_{i}(T/_{i}X) \quad (< T) \quad (1.8)
\]

The index \( i \) can be thought of as a “mode variable”: in other words, the two slashes of a type-raised category may carry any mode, as long as they are the same mode.

Baldridge’s goal for MMCCG is to provide a handle on the full power of CCG, allowing us to harness all of it when truly necessary, while limiting its availability when it is not:

\[
\begin{array}{cccc}
\text{strong} & \text{with milk} & \text{coffee} \\
N/N & (N_{\ast}\backslash N)_{\ast}N & N \\
& N\backslash \ast N & N_{\ast}(N/\ast N) \\
\end{array}
\]


1.8 Extra-combinatory rules

Certain rules which occur in CCG grammars are either not motivated by the CCG combinators (B, T, S), or contravene one or more of the stipulations on combinatory rules of Section 1.6.6. Nonetheless, for reasons of efficiency, practicality, or increased coverage, the case for extra-combinatory rules has long been made in the literature, and they are a common part of CCGbank-style wide-coverage grammars.

1.8.1 Rules unconstrained by the stipulations on CCG rules

Steedman (2000) recognised that the Principle of Consistency (Principle 2) rules out order-changing type-change rules of the form:

\[
X \rightarrow T/(T/X)
\]
\[
X \rightarrow T\backslash (T\backslash X)
\]

because these allow arguments to be consumed in a way which contravenes the directionality of the head’s slash. Leaving the rules unconstrained has the effect of obliterating the
directionality distinctions made in the lexicon — an undesirable state of affairs in configurational languages.

\[
(38) \quad \frac{C}{B/(B/C)} \quad \frac{B/C}{B/(B/C)} \quad \frac{B}{B/(B/C)}
\]

However, such rules are *necessary* to analyse left- or right-dislocation, which is by definition the movement of a constituent to the left or right periphery of a sentence. To control the overgeneration resulting from order-changing type-change rules, Steedman (1987) proposed that in languages where dislocation is licensed, rules such as \( X \rightarrow T/(T/X) \) be licensed only when the input category is adjacent to the left periphery of the sentence, while Steedman (2000) suggests that because topicalisation is typically indicated in spoken English by special intonation, or in the written language by setting apart the dislocated constituent with a comma, the rule can be avoided by providing additional lexical categories for heads which can undergo dislocation.

For English CCGbank, Hockenmaier and Steedman (2005) selected the second analysis through lexical ambiguity, arguing that the topicalisation constructions which manifest as left-dislocation in English are sufficiently rare in the Penn Treebank that the analysis through lexical ambiguity suffices. Later, in Chapter 2, we argue for the opposite position in Chinese, to motivate our use of the order-changing type-change rule in Steedman (1987).

### 1.8.2 Absorption rules

Absorption rules emulate the lexically-specified derivational control provided by multimodal CCG (MMCCG). MMCCG allows individual lexicon entries to specify which of the combinatory rules is licensed for any given slash (Baldridge, 2002). Consider the sentence-final period, which in English CCGbank is analysed through the extra-combinatory rule which *absorbs* the punctuation token:

\[
X . \rightarrow X
\]  

(1.9)

An alternate, pure-ccg analysis of the sentence-final period as \( \vdash S[decl] \backslash S[decl] \) overgenerates when the period is allowed to compose with other post-sentential modifiers such as sentential adverbs:
1.9. The trade-off between derivational and lexical ambiguity

In the English reduced relative clause (RRC) construction, the relative pronoun that can be omitted in an object-gapped relative clause.\(^\text{14}\)

\begin{align*}
(40) \quad \text{a.} & \quad \text{This is the bunny, that Dave sent } t_i. \\
\text{b.} & \quad \text{This is the bunny, Dave sent } t_i.
\end{align*}

The usual CCG analysis of the relative clause is “powered by” the category of the relative pronoun that, in the sense that its category collects the arguments on either side which make up the relative clause:

\(^{14}\) An object-gapped relative clause is one whose main verb’s object argument has been extracted. For instance, in both sentences in (40), the syntactic object and semantic patient of sent is a gap \(t_i\) — an unrealised constituent.
A problem with an analysis “powered by” a particular lexical item, is how to maintain the analysis when that lexical item is not realised. The more general phenomenon of syntactic form (how a constituent appears) being at odds with semantic form (how a constituent behaves) arises time and time again in CCG analyses of language fragments, and also informs the decisions that we made in developing Chinese CCGbank. We discuss three mechanisms from the literature for handling these form-function discrepancies.

1.9.1 Analyses through lexical ambiguity

A possible analysis for the RRC construction is that if the lexical item powering the construction is not present, perhaps another lexical item which participates in the construction can power it instead. We refer to this solution, which proposes additional categories for lexical items when they participate in special constructions, as the analysis through lexical ambiguity.

For the RRC construction, Honnibal (2010) presents two possible analyses through lexical ambiguity: one powered by the RRC head (42-a), and one by the RRC verb (42-b).

\[
\begin{array}{c|c|c|c|c|c}
\text{the } & \text{bunny} & \text{that} & \text{Dave} & \text{sent} \\
\hline
\text{NP[bp]/N} & \text{NP} & (\text{NP} \rightarrow (\text{S[\text{dcl}]}/ \text{NP})) & (\text{NP} \rightarrow (\text{S[\text{dcl}]}/ \text{NP})) & (\text{S[\text{dcl}]}/ \text{NP})/ \text{NP} \\
\hline
\end{array}
\]

\[
\begin{array}{c|c|c|c|c|c}
\text{NP} & \text{NP} & (\text{NP} \rightarrow (\text{S[\text{dcl}]}/ \text{NP})) & (\text{NP} \rightarrow (\text{S[\text{dcl}]}/ \text{NP})) & (\text{NP} \rightarrow (\text{S[\text{dcl}]}/ \text{NP})) & (\text{NP} \rightarrow (\text{S[\text{dcl}]}/ \text{NP})) \\
\hline
\end{array}
\]

\[
\begin{array}{c|c|c|c|c|c}
\text{NP} & \text{NP} & (\text{NP} \rightarrow (\text{S[\text{dcl}]}/ \text{NP})) & (\text{NP} \rightarrow (\text{S[\text{dcl}]}/ \text{NP})) & (\text{NP} \rightarrow (\text{S[\text{dcl}]}/ \text{NP})) & (\text{NP} \rightarrow (\text{S[\text{dcl}]}/ \text{NP})) \\
\hline
\end{array}
\]
Analyses through lexical ambiguity have the advantage that the set of lexical items licensed for special syntax can be specified in the lexicon. However, if the language's grammar licenses an open class of lexical item, two weaknesses come to the fore. Concretely, considering the rrc analysis in (42-a) powered by the noun:

1) The lexicon must contain an rrc category of the form \( NP/(S[dcl]/NP) \) for every noun which is a candidate head of the rrc construction (lexical sparsity).

2) Modifiers of bare nouns, which canonically have the shape \( N/\ast \), must additionally have the shape \( (NP/(S/NP))/\ast \) (modifier proliferation).

Analogous weaknesses also exist in the analysis powered by the verbal category.

### 1.9.2 Analyses through derivational ambiguity

In the rrc construction, the function of the constituent Dave sent is a noun post-modifier \( (NP\backslash NP) \). However, its form is an object-gapped sentence \( (S[dcl]/NP) \). When the relative pronoun is overt, its category \( (NP\backslash NP)/(S[dcl]/NP) \) acts as a functor from form to function, mediating the conversion between the two.

Thus, a natural extension to handle rrc is to reintroduce an extra-combinatory unary type-change rule which performs this conversion:

\[
S[dcl]/NP \rightarrow NP\backslash NP
\]  

(1.10)

This is the solution of Hockenmaier and Steedman (2005) for English CCGbank, which we refer to as the analysis through derivational ambiguity. A disadvantage of relying on derivational ambiguity is that the semantics are no longer determined by the theory, since the theory does not license arbitrary rewrite rules of the form of Rule 1.10.

Aone and Wittenburg (1990) interpret unary rules as the combinatory analogue of zero morphemes, observing that a rule such as \( S[dcl]/NP \rightarrow NP\backslash NP \) exactly corresponds to the operation of an overt functor \( (NP\backslash NP)/(S[dcl]/NP) \). Unary rules can thus be considered as a means of analysing zero morphemes in a ccg parser, and the associated semantics of the unary rule can be derived from the semantics of the overt functor.

The tug-of-war between lexical ambiguity and derivational ambiguity — two means of analysing the same class of form-function distinctions — looms large in this work and the ccg literature in general.
1.9.3 Hat categories

Honnibal (2010) observed that English phrases headed by a gerund have VP form but NP function:

(43)  a. I gave doing things his way a chance.
   b. Doing things his way gave me a chance.

Because CCG categories encode syntactic form, the same trade-offs encountered in analyses through derivational ambiguity and lexical ambiguity arise, leading either to a multiplicity of verbal categories mentioning a gerund-form sentence $S[nom]$:

\[
gave \vdash ((S[dcl] \backslash S[nom]) / NP) / NP
\]

or, if gerund-form sentences are given NP function, represented as $NP[ger]$, a multiplicity of noun modifier categories mentioning gerund-form NPs:

\[
\text{quickly} \vdash (S \backslash NP) \backslash (S \backslash NP)
\]

\[
\vdash NP[ger] \backslash NP[ger]
\]

The solution of Honnibal (2010) was to compile type-change rules into category structure, so that the applicability of type-change rules could be specified lexically, in the same way that multi-modal CCG allows the applicability of combinatory rules to be specified lexically. This is achieved by augmenting category structure with a category-valued field called hat, written as a superscript:

\[
doing \vdash (S[dcl] \backslash NP)^{\hat{NP}} / NP
\]

The hat annotation means that at some point during the derivation when the outermost category has a hat value, the hatted category may be discharged by applying the unhat rule (H), which simply replaces the hatted category with the category inside its hat:

\[
X^\hat{H} \rightarrow H \quad (H)
\]

This solution ameliorates the category proliferation issues arising from the analyses through lexical ambiguity, and pushes the specification of type-change rules back into the
lexicon, at the cost of increasing the size of the lexicon. As we demonstrate for Chinese, form-function ambiguities arise in a number of contexts, including the zero copula construction. We will note in our analysis where a solution through hat-ccg is especially perspicuous, but in general, hat-ccg offers an alternative analysis to any type-change rule by compiling it into the lexicon.

1.10 Notational conveniences for categories

We introduce notational conveniences which are useful when describing categories, or generalising over sets of categories. We sometimes wish to underspecify the directionality of a slash. The bar | stands for either a forward or a backward slash, so that NP|NP indicates either NP/NP or NP\NP.

We denote an assignment of a category C to a lexical item w by writing w ⊢ C. For instance, quickly ⊢ (S\NP)/(S\NP) pairs the lexical item quickly with a category.

Due to their prevalence, we introduce abbreviations which coincide with the definitions of intransitive, transitive and ditransitive verbs:

\[
\begin{align*}
VP & \equiv S[\text{dcl}]\backslash NP. \\
TV & \equiv VP/NP. \\
DTV & \equiv TV/NP.
\end{align*}
\]

We will also write, e.g. \(VP[q]\) to mean \(S[q]\backslash NP\).

We adopt the Steedman (2000) “$\$-schematisation”, which allows for the specification of an entire (infinite) family of categories with the same result category. Given a category C, the set of categories C\$ consists of all functors with the result category C. For instance, \((S[\text{dcl}]\backslash NP)\$\) compactly specifies all categories into the predicate category \(S[\text{dcl}]\backslash NP\), including \((S[\text{dcl}]\backslash NP)/NP\), \((S[\text{dcl}]\backslash NP)/PP\), and so on.

We can also restrict the $\$-schematisation to rightward or leftward functors — \(C/\$\) denotes all rightward functors into \(C\), and \(C/\$\) denotes all leftward functors into \(C\).

If the $\$ has an index, assume that coindexed $\$ signs refer to the same argument stack. For instance, \(NP/\$\) contains the category \((NP/\$)/(NP/\$)\), since both coindexed $\$ signs stand in for the argument S, but not \((NP/(S/\$))/\$\).

\[15\]Despite the notation, the phrase/terminal distinction of phrase structure grammar does not exist in ccg. Thus, VP represents any span whose category is \(S[\text{dcl}]\backslash NP\).
Finally, we introduce a “*-schematisation” to specify a family of modifier categories. In this convention, $X/\ast$ (respectively $X\backslash\ast$) represents the set of all rightward (or leftward) modifier categories into $X$.

For example, $N/\ast$ represents the set:

$$\{N/N, (N/N)/(N/N), ((N/N)/(N/N))/((N/N)/(N/N)), \ldots\}$$

The interpretation of coindexed $\ast$-signs is analogous to that of coindexed $\$-signs.

### 1.11 Applications of CCG

Its transparent syntax-semantics interface, simple and consistent accounts of coordination phenomena and other constructions, the availability of efficient parsing algorithms, wide-coverage grammars, and practical parsers have made CCG an attractive basis for many NLP systems, including generation, semantic role labelling, question answering and machine translation.

#### 1.11.1 Generation and realisation

In the black-box conception of parsing as a mapping from strings of a language to logical forms, surface realisation is the inverse mapping from a logical form to a natural language string. Except for systems such as Ratnaparkhi (2000) which map directly from logical forms to surface strings, most realisation systems employ some grammar formalism with a syntax-semantics interface to decompose the task into converting the logical form into structures of the grammar formalism (for instance, LFG f-structures, CCG categories or HPSG attribute-value matrices), then searching for likely candidate paraphrases given those structures.

The type-driven analysis of coordination phenomena in CCG which assigns principled types to non-constituent types such as argument clusters, right-node-raised conjuncts and other gapped phrases led White and Baldridge (2003) to use CCG to constrain the space of possible syntactic realisations in a generation system. In this system, logical forms represented as expressions from hybrid logic dependency semantics (HLDS) are associated with pairs of lexical items and CCG categories, and chart generation is used to produce candidate realisations whose logical form covers the terms of the input HLDS expression.

Further work in CCG realisation focused on extracting the generation lexicon from a wide-coverage lexicon, avoiding the onerous task of manual annotation. White et al. (2007)
extracted logical forms from the predicate-argument structure of English CCGbank. Espinosa et al. (2008) then showed that the supertagging approach which Clark and Curran (2004b) had previously shown to greatly increase both efficiency and accuracy in tagging lexical items with CCG categories could also be used to tag logical forms with CCG categories in preparation for chart realisation.

1.11.2 Supertagging

Joshi and Srinivas (1994) proposed that an approach called supertagging could greatly reduce the ambiguity encountered by a polynomial-time parser, by pre-applying a linear-time tagging process which, for each input lexical item, assigns a structurally rich description (a supertag). Based on the observation that many lexical dependencies can be assigned given only local context, supertagging “almost parses” the input, allowing the parser to consider a smaller and more focused space of decisions.

While the “rich descriptions” explored by Joshi and Srinivas (1994) were the elementary trees of Lexicalised Tree Adjoining Grammar (LTAG; Schabes et al., 1988), Clark (2002) showed that supertagging was also effective at assigning CCG categories, introducing the \( \beta \) mechanism which the C&C parser uses to control the degree of supertagger ambiguity. In a simple extension from 1-best tagging to multitagging, a supertagger can assign all categories whose probability is within a factor of \( \beta \) of the highest-confidence category.

Using \( \beta \) as a handle on ambiguity, Clark and Curran (2004a) developed adaptive supertagging, in which the supertagger iteratively increases ambiguity levels until the parser can obtain a spanning analysis with a given assignment of categories.

CCG supertags have been used as features in a semantic role labelling system (Boxwell et al., 2009), supplementing the sparser tree-path features typically used in SRL. Birch et al. (2007) also applied supertags as features in a factored translation model, improving the linguistic plausibility of reorderings by incorporating sequences of supertags as features.

1.11.3 Question answering

Question answering (QA) is the retrieval of answers from a knowledge base, in response to user queries phrased as natural-language questions.

Clark et al. (2004) observed that a supertagging model acquired purely from WSJ text performs very poorly in the QA domain, because \( \text{wh} \)-questions are so rare in CCGbank that attestations of certain types are rare (how, which), or even completely absent (e.g. what + N). They annotated 1171 What or What + N questions only with their correct CCG categor-
ies, without providing manually annotated complete CCG derivations. They retrained the supertagger, increasing word-level accuracy from 84.8% to 98.1% on their question-only dataset. Versions of C&C exploiting this improved supertagging model have subsequently been incorporated into several TREC question answering systems (Ahn et al., 2004, 2005).

Zettlemoyer and Collins (2005) exploited the transparent syntax-semantics interface of CCG to acquire a CCG grammar from GeoQuery data which transformed input queries to logical form.

1.12 Summary

In this chapter, we have described Combinatory Categorial Grammar, a lexicalised formalism whose transparent syntax-semantics interface endows it with clean, type-driven analyses of a range of constructions including extraction, as well as coordination phenomena including non-constituent coordination and the argument cluster coordination. We have traced its descent from the pure categorial grammars of Ajdukiewicz and Bar-Hillel, through to Steedman’s CCG, which supplemented the AB categorial grammar with combinatory rules, various augmentations of this base formalism, and the style of CCG grammar developed by Hockenmaier for the influential English CCGbank, which in turn influences the development of our Chinese CCGbank.

We are now ready to present the first contribution of this thesis: an analysis of Chinese syntax through the lens of Combinatory Categorial Grammar.
Chapter 2

The elements of Chinese ccg

Only by comparison can we discern the commonalities and differences in linguistic representation between languages.

Lü Shuxiang (呂叔湘; 1904–1998)

The contribution of this and the following chapter is the first analysis of the distinctive aspects of Chinese syntax through the lens of Combinatory Categorial Grammar.

We present a Chinese ccg grammar which accounts for its canonical argument-taking behaviour, coordination phenomena, the handling of punctuation, and the analysis of verb compounding strategies. We defer the analysis of syntactic phenomena which induce non-local dependencies and other complex effects to the following chapter.

ccg analyses of languages, including fragments of Turkish (Çakıcı, 2005), German (Hockenmaier, 2006), Arabic (Boxwell and Brew, 2010) and Italian (Bos et al., 2009), have shown that ccg is capable of handling cross-linguistic syntax in the practical setting. These accounts of cross-linguistic syntax have supplemented the literature with generic analyses of free word order and scrambling effects, pro-drop, topicalisation and null-copula constructions, which can be brought to bear on future ccg analyses in new languages.

Bender (2009) argues that a future conception of language-independent nlp could involve decomposing a language into its typological characteristics, and then selecting cl techniques, resources or analyses accordingly.

One strategy for unifying analyses for cross-linguistic syntax for grammar engineering is exemplified by the ParGram project for lfg. The work which resulted in the English, French and German ParGram corpora used a top-down strategy, in which the lfg analyses
for common constructions in the three languages were harmonised so that they would share similar $f$-structures (Butt et al., 1999). This similarity can be leveraged for machine translation, with the common $f$-structure being used as the intermediate representation in interlingual machine translation (Frank, 1999).

Research in CCG grammar engineering, unlike the ParGram program for LFG, has proceeded in a decentralised manner. As CCG analyses appear for additional languages, the literature is enriched with analyses of constructions which may be incorporated into other languages which exhibit the same kinds of syntax. To stress the reusability of the CCG analyses developed in this chapter, we cross-reference each fragment of the CCG grammar against its corresponding feature value in the World Atlas of Language Structures (WALS; Haspelmath et al., 2008).

This chapter will deal with our Chinese CCG grammar in the abstract — without reference to a means of extracting the grammar, or a particular instantiation of the grammar. We isolate the grammar from its instantiation, because the goals of the abstract and concrete grammar differ. We design the concrete Chinese CCGbank grammar for the end goal of serving as the input to a wide-coverage statistical CCG parser, motivating tradeoffs between formalistic purity and efficiency. We examine some of these tradeoffs in Chapter 6. By contrast, the goal of the abstract Chinese CCG grammar is to reconcile analyses of Chinese syntax which already exist in the linguistics literature with the CCG setting. While the two grammars mostly overlap, neither entirely contains the other.

We begin by characterising the canonical word order typology of Chinese. We then consider head-modifier order of simple phrases, before considering the larger units of coordination and sentence-level syntax. Finally, we consider various verb compounding strategies available in Chinese, concluding that certain compounding strategies should be considered morphological operations opaque to syntax.

### 2.1 A review of CCG grammars in the literature

The first wide-coverage CCG corpus in the literature was the English CCGbank developed by Hockenmaier (2003), who implemented an automatic corpus conversion from the Penn Treebank (Marcus et al., 1994) which projected PTB syntactic analyses into CCG derivations. The development of English CCGbank directly enabled a line of research into statistical CCG parsing, leading to work in CCG parsing with generative models (Hockenmaier, 2001), discriminative chart parsing (Clark and Curran, 2003) and transition-based shift-reduce parsing (Zhang and Clark, 2003).
2.1. **A review of CCG grammars in the literature**

The grammar implemented by Hockenmaier (2003) for English CCGbank incorporated analyses which were current in the CCG literature, projecting the Penn Treebank annotations into these standard CCG analyses. The constructions analysed include the standard analyses for extraction and non-constituent coordination phenomena from Steedman (2000), parasitic gap phenomena from Szabolcsi (1989), and pied piping, tough-movement and heavy NP shift from Steedman (1996).

Cha et al. (2002) observed that the agglutinative nature of Korean morphology justifies a categorial lexicon of morphemes rather than words, and developed a statistical model and chart parsing method suitable for the free word order characteristics of Korean.

Hockenmaier (2006) also developed a corpus conversion of the German dependency grammar-based corpus TIGER (Brants et al., 2002) to CCG, with a grammar which captured the main clause/subordinate clause word order characteristics of German. However, the grammar calls for a large degree of lexical ambiguity on verbs, requiring separate categories for main clause, subordinate clause, and scrambled frames. One solution Hockenmaier (2006) suggests is the approach of Hoffman (1996), who proposed the multiset-CCG extension to account for the flexibility in argument-taking order seen in languages such as Turkish, Japanese and Korean.

Bos et al. (2009) extracted a wide-coverage CCG grammar from the Turin University Treebank, whose native annotation style, like that of the German TIGER corpus, is defined over dependency structures rather than the constituent trees of Penn Treebank-style corpora. Bos et al. (2009) also reject the CCGbank-style absorption analyses of punctuation in favour of a category-driven analysis in which punctuation tokens act as functors, allowing the grammar to capture the pairing behaviour of matched punctuation tokens.

Each of these grammars provided the CCG literature with analyses for new objects of syntax. Just as we adapt existing analyses from the literature to analyse aspects of Chinese syntax, we anticipate that future CCG corpora can adapt our analyses to new languages and new CCGbanks.
Chapter 2. The elements of Chinese ccg

2.2 On developing the Chinese CCGbank grammar

At the time Hockenmaier (2003) began the development of English CCGbank, ccg parsers such as the one in Grok (Baldridge, 2002) were available, but were built for relatively small grammars, although Hockenmaier et al. (2004) describes an effort to merge CCGbank with a curated ccg grammar for use in the Grok system. A decade later, wide-coverage ccg parsing is well-developed (Clark and Curran, 2007b), and the generate-train-evaluate cycle can be used to inform our grammar engineering decisions.

While we present this work thematically as an abstract analysis of Chinese ccg (Chapters 2 and 3) followed by a description of how we extract an approximation to that analysis from the Penn Chinese Treebank (Chapters 4 and 5), this structure belies the true development process of Chinese CCGbank. The development process began with a close reading of the Penn Chinese Treebank syntactic annotation guidelines (Xue et al., 2000), POS tagging guidelines (Xia, 2000a) and segmentation guidelines (Xia, 2000b), through which we determined which constructions were handled in the 

<table>
<thead>
<tr>
<th>Work</th>
<th>Language</th>
<th>Wide-coverage</th>
<th>Type-change rules</th>
<th>Conversion</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hoffman (1996)</td>
<td>Turkish</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Argument scrambling</td>
</tr>
<tr>
<td>Çakıcı (2008)</td>
<td>Turkish</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Morpheme-level categories; pro-drop</td>
</tr>
<tr>
<td>Cha et al. (2002)</td>
<td>Korean</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Morpheme-level categories</td>
</tr>
<tr>
<td>Hockenmaier (2003)</td>
<td>English</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Scrambling; V2 word order</td>
</tr>
<tr>
<td>Hockenmaier (2006)</td>
<td>German</td>
<td>Yes</td>
<td>-</td>
<td>Yes</td>
<td>Scrambling; V2 word order</td>
</tr>
<tr>
<td>Bos et al. (2009)</td>
<td>Italian</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Categorial punctuation</td>
</tr>
<tr>
<td>Boxwell and Brew (2010)</td>
<td>Arabic</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>pro-drop</td>
</tr>
</tbody>
</table>

Table 2.1: Language fragments analysed through ccg

We generated many iterations of Chinese CCGbank, computing measures of ambiguity such as categories per token (average lexical ambiguity), lexicon size and the lexicon’s category and rule growth characteristics. Because the computational complexity of cky-style parsing $O(|R| \cdot n^3)$ contains a factor $|R|$ representing the number of possible rule instanti-
2.2. **On developing the Chinese CCGbank grammar**

On developing the Chinese CCGbank grammar (Bangalore and Joshi, 2010), the number of combinatory rule instances attested in the lexicon has an impact on performance.

During development, we also considered the relative frequency of each construction in the PCTB, generally attacking constructions in order of their frequency in the source corpus. Where relevant, we will quantify the prevalence of a construction when discussing its CCG analysis in this and the following chapter. Whether a construction is frequent can affect how we choose to analyse it, as we discuss later in the case of S NP apposition or topicalisation.

About halfway through the course of its development, we turned our attention to modifying the C&C parser, which demonstrated state-of-the-art performance for dependency recovery trained on English CCGbank, to accept Chinese CCGbank as training and evaluation data. This enabled a new methodology for grammar engineering called *parser-aided corpus conversion*, which we present and explore in Chapter 7. In parser-aided corpus conversion, the generate-train-evaluate cycle can be used to quantify the impact of particular annotation decisions in a CCGbank. The use of such a feedback loop in grammar engineering is reminiscent of the approach of King et al. (2000) in XLE and Bender et al. (2008) in LinGO Redwoods, in the context of hand-engineered wide-coverage grammars. They suggest incorporating a parser into the grammar engineering loop, to quantify the increase in ambiguity caused by a particular annotation decision, or to evaluate the coverage of a chosen analysis over a corpus. Our methodology for selecting analyses for Chinese CCGbank involves training full statistical models trained on candidate versions of Chinese CCGbank. We choose analyses which balance lexicon size, parser ambiguity, accuracy and coverage, as informed by evaluation over the resulting statistical models.

Evaluating on a wide-coverage parser can uncover different and unexpected properties of the annotation, such as interactions between unary rules leading to undesirable ambiguity. Since the end goal of this work is to infer efficient wide-coverage CCG parsing models for Chinese, these constraints feed back into the corpus conversion and may lead us to avoid analyses which lead to poor practical performance.

---

1 The time complexity of naïve CKY-style generalised CCG parsing is exponential time: subtrees derived in different ways can no longer share a chart entry. Vijay-Shanker and Weir (1993) presented an algorithm which applies structure-sharing to the stack structure of CCG categories, yielding $O(n^2)$ worst-case time complexity.

However, the “seen rules restriction” (used by the parser of Hockenmaier (2003) and also C&C during parsing (Clark and Curran, 2007b)), restricts the grammar to strong context-freeness (Fowler and Penn, 2010). The time complexity of the recognition algorithm in these parsers is hence the same as CKY over context-free grammars.
Chapter 2. The elements of Chinese CCG

The resources available to us shaped the Chinese CCGbank development methodology. For instance, because the C&C parser, the target for Chinese CCGbank, was built for and is optimised for CCGbank-style grammars, characteristics of the Chinese CCGbank annotation, such as type-change punctuation rules, the feature convention, the encoding of dependencies through head variable annotations, follow those of English CCGbank.

The second constraint on Chinese CCGbank is the minimisation of manual annotation effort. Some of the analyses we present in this and the following section, such as the analysis of separable V-O compounds in Section 2.5.2, would be difficult to extract without additional annotation or lexical resources.

The final constraint on Chinese CCGbank was the annotation style of its source corpus, the Penn Chinese Treebank. Xue et al. (2005) describe three trade-offs — speed, quality and usefulness — which informed the compromises made during the development of the PCTB. When the PCTB intentionally under-specifies a dimension of the annotation for annotator efficiency, the compromise affects Chinese CCGbank too.

2.3 Basic sentence typology and position of adjuncts

The analysis in this section begins our account of the syntax of Chinese through CCG. The goal of this and the following chapter is to demonstrate that a succinct account of Chinese syntax can be achieved using the machinery of CCG, harmonising analyses from the Chinese generative syntax literature with analyses in the CCG literature.

We start by analysing the basic, unmarked word order of subjects and predicates, and the syntactic distribution of PP complements and adjuncts.

2.3.1 Canonical word order

Like English, Chinese is fundamentally an isolating subject-verb-object (svo) language, so that the canonical word order and modifier-head order in simple sentences may resemble that of English:

(1) 西藏 金融 工作 取得 显著 成绩
Tibet banking project achieve remarkable performance
Banking project in Tibet achieves remarkable performance

Hence, the shape and directionality of Chinese verbal categories resembles those of English too (Table 2.2).
### Table 2.2: Basic verbal categories in Chinese and English

<table>
<thead>
<tr>
<th>Chinese example</th>
<th>Category</th>
<th>Verb type</th>
</tr>
</thead>
<tbody>
<tr>
<td>跑 run</td>
<td>$S[\text{dcl}]\backslash NP$</td>
<td>Intransitive verb</td>
</tr>
<tr>
<td>带 wear</td>
<td>$(S[\text{dcl}]\backslash NP)/NP$</td>
<td>Transitive verb</td>
</tr>
<tr>
<td>给 give</td>
<td>$((S[\text{dcl}]\backslash NP)/NP)/NP$</td>
<td>Ditransitive verb</td>
</tr>
</tbody>
</table>

However, Chinese adjunct phrases precede the verb, while English adjunct phrases may precede or follow the verb. As such, Chinese VP adjuncts only have the category $VP/VP$, while English VP adjuncts can additionally have $VP\backslash VP$. Chinese PPs which occur post-verbally are verbal complements rather than adjuncts.

(2)  

a. I will buy vegetables tomorrow.

<table>
<thead>
<tr>
<th>I</th>
<th>tomorrow</th>
<th>buy</th>
<th>vegetables</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td>$(S\backslash NP)/(S\backslash NP)$</td>
<td>$(S[\text{dcl}]\backslash NP)/NP$</td>
<td>NP</td>
</tr>
<tr>
<td></td>
<td>$S[\text{dcl}]\backslash NP$</td>
<td>$S[\text{dcl}]\backslash NP$</td>
<td>$S[\text{dcl}]$</td>
</tr>
</tbody>
</table>

b. I buy vegetables at the supermarket.

<table>
<thead>
<tr>
<th>I</th>
<th>at</th>
<th>supermarket</th>
<th>buy</th>
<th>vegetables</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td>$(S\backslash NP)/(S\backslash NP)/NP$</td>
<td>$(S[\text{dcl}]\backslash NP)/NP$</td>
<td>NP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S[\text{dcl}]\backslash NP$</td>
<td>$S[\text{dcl}]\backslash NP$</td>
<td>$S[\text{dcl}]$</td>
<td></td>
</tr>
</tbody>
</table>

Because PP adjuncts take a distinguished pre-verbal position, one particular PP attachment ambiguity present in English is notably absent in Chinese:

(3)  

a. 我用望远镜看见那个女孩子

1sg use telescope see-achieve:res that mw girl

I see that girl by means of a telescope.

b. 我看见那个用望远镜的女孩

1sg see-achieve:res that mw use telescope de girl

I see that girl who is using a telescope.
Chapter 2. The elements of Chinese CCG

### Table 2.3: Subcategorisation frames in the PCTB, truncated at frequency 100

<table>
<thead>
<tr>
<th>Number of types</th>
<th>Frame</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>9381</td>
<td>V</td>
<td>大; 多; 好; 新; 高</td>
</tr>
<tr>
<td>5883</td>
<td>V NP</td>
<td>是; 有; 进行; 为; 成为</td>
</tr>
<tr>
<td>1201</td>
<td>V IP</td>
<td>说; 有; 表示; 认为</td>
</tr>
<tr>
<td>694</td>
<td>V QP</td>
<td>达; 增长; 为; 占; 有</td>
</tr>
<tr>
<td>461</td>
<td>V NP IP</td>
<td>让; 使; 前; 成为</td>
</tr>
<tr>
<td>436</td>
<td>V PP</td>
<td>是; 位; 用; 发生; 设</td>
</tr>
<tr>
<td>384</td>
<td>V V</td>
<td>看; 找; 做; 未; 恕</td>
</tr>
<tr>
<td>339</td>
<td>V VP</td>
<td>是; 要; 会; 能; 可以</td>
</tr>
<tr>
<td>212</td>
<td>V NP QP</td>
<td>完成; 实现; 利用; 有; 批准</td>
</tr>
<tr>
<td>171</td>
<td>V LCP</td>
<td>在; 包括; 占; 到; 达</td>
</tr>
<tr>
<td>133</td>
<td>V NP NP</td>
<td>给; 占; 给予; 判处; 提供</td>
</tr>
</tbody>
</table>

2.3.2 Subcategorisation in Chinese

Although the earliest categorial grammars only distinguished two or three atoms — S (type t), NP (type e) and N (type ⟨e, t⟩) — Hockenmaier (2003) proposes the additional atom PP in order to account for English words which subcategorise specifically for PP complements, such as the verb increase or the adverb according:

(4) a. Stocks [ increased ](S|dcl|\NP)/PP [ by 30% overnight ]\PP  
    b. [ According ](VP/VP)/PP [ to authorities ]\PP

To determine the atom inventory for the PCTB, we examined the distribution of subcategorisation frames in the corpus. Table 2.3 shows the subcategorisation frames of all verbs in the PCTB down to a frequency of 100, with examples for each subcategorisation frame sorted by frequency. Because sufficiently many verbs subcategorise specifically for QP (quantifier phrase — a NP modified by a numeral) and LCP (localiser phrase) complements, following the same argumentation in Hockenmaier (2003) to justify the atom PP, we augment the atom inventory of Chinese CCGbank to include the atoms QP and LCP.
2.4 Constructions headed by NP

The distinctive measure words of Chinese, which appear when noun phrases are modified by a determiner or by a numeral, sometimes reflect the physical shape of the head noun (一条裤子 one pair of pants, 一条电线 one wire), or select a particular configuration of the head noun (一本书 one book vs. 一箱书 one box of books). We consider some analyses of NP internal structure from the Chinese syntax literature, and argue for their representation in CCG.

2.4.1 Quantifier phrases and measure words

Any quantification of a Chinese noun, whether by a numeral (e.g. 三十三 thirty-three) or by a quantifier (e.g. 每 each), obligatorily involves a measure word, which comes between the quantifier and the noun:

(5) a. 三十三头牛
   thirty-three mw cattle
   33 head of cattle

b. 每个学生
   each mw student
   each student

The measure word is selected for by the head noun. Some measure words group together nouns with similar physical attributes (e.g. 条 tiao, a measure word for long, slender objects), while some are effectively only used with a single noun (e.g. 匹 mw for horses). The choice of measure word can also disambiguate different senses of the same head noun:

(6) a. 两门课
   two mw lesson
   two courses (for instance, “Introduction to AI” and “Algorithms II”)

b. 两节课
   two mw lesson
   two periods (for instance, a morning lecture and an afternoon lecture)

Thus, the quantifier phrase specifies a quantity of its head noun, and contains an obligatory measure word. In the analysis of Huang et al. (2008), quantifier phrases (NumP in the text) are considered a phrasal projection of a numeral head which takes a ClP (classifier phrase) complement:
three people

However, while this analysis captures the fact that the measure word places selectional restrictions on the head of its complement NP, it hypothesises a Cl + NP unit which fails the coordination constituency test:

\( \text{三匹马或者一头牛} \)

three horse: mw horse or cow: mw cow

(intended) three horses or cows

On the other hand, the unit Num + Cl forms more natural conjuncts:

\( \text{二百五十张或者一令白纸} \)

250 sheet: mw or one ream: mw paper

250 sheets or one ream of paper

In order to capture the fact that the measure word is obligatory, this unit Num + Cl must be headed by the numeral, taking the measure word as an argument. Assigning the tentative category \( X \) to the measure word, the category of the numeral 250 would have the form:

\[ 250 \vdash (NP/NP)/X \]

Finally, we consider what category should be assigned to the measure word. Diachronically, many measure words derive from nouns. However, the process which generates measure words from nouns is no longer productive in the modern language:

\( \text{三瓶啤酒} \)

three mw beer

three bottles of beer

\( \text{三瓶健力士} \)

three mw Guinness

*three beers of Guinness
Furthermore, measure words can be reduplicated to obtain a distributive reading (Yip, 2008), while nouns cannot:

\[
\begin{align*}
\text{(11) a. } & \quad \text{朵 朵 花} \\
& \quad \text{Mw Mw flower}
\text{b. } & \quad *\text{花 花 玫瑰} \\
& \quad \text{Mw Mw rose}
\end{align*}
\]

Unlike nouns, measure words accept only a small set of modifiers.

\[
\begin{align*}
\text{(12) a. } & \quad \text{手头 有了 大 把 活 钱} \\
& \quad \text{at.hand have asp big Mw living money} \\
& \quad \text{He has a great deal of money at hand.}
\text{b. } & \quad \text{一 整 套 基本政策} \\
& \quad \text{one whole Mw basic policy} \\
& \quad \text{a whole set of basic policies}
\text{c. } & \quad *\text{那 红 头} \\
& \quad \text{that red Mw} \\
& \quad *\text{that red (one)}
\end{align*}
\]

Lastly, with the exception of units of measurement, measure words are largely monosyllabic, which contrasts with the prosodic distribution of the modern Chinese noun. On the basis of their lexical distinctiveness and modifier behaviour, we adopt the additional atom $M$ for Chinese, so that the final category of a numeral in our analysis is $(NP/NP)/M$, resulting in the following structure for $NP$:

\[
\begin{align*}
(\text{13) } & \quad \text{三 人}\ 	ext{three people}
\end{align*}
\]

A number of classifiers, including 天 day: Mw, 年 year: Mw, 元 yuan, do not form units with NPs. As these are measure words of measurement (定量量词), the Num + Cl compound
in these cases indicates an extent or quantity rather than modifying a noun phrase. We analyse compounds of these classifiers as resulting in a \( QP \) rather than an \( NP \):

(14) will reach US$2.5 billion

\[
\begin{tikzpicture}
  \node (S) at (0,0) {\( S[dcl]NP \)};
  \node (NP) at (1,-1) {\( (S\ NP)/(S\ NP) \)};
  \node (S1) at (2,-2) {\( S[dcl]NP \)};
  \node (QP) at (3,-3) {\( (S[dcl]NP)/QP \)};
  \node (reach) at (4,-4) {reach};
  \node (QP/M) at (5,-5) {QP/M};
  \node (M) at (6,-6) {M};
  \node (USD) at (7,-7) {USD};
  \draw (S) -- (NP);
  \draw (NP) -- (S1);
  \draw (S1) -- (QP);
  \draw (QP) -- (reach);
  \draw (reach) -- (QP/M);
  \draw (QP/M) -- (M);
  \draw (M) -- (USD);
\end{tikzpicture}
\]

**Elision of numeral in \( \text{Num} + \text{M} \) construction**

A measure word must intervene when a quantity modifies a noun. However, when the quantity in question is \( \text{one} \), and the measure word is concrete (as in \( \text{one sentence} \)), the quantity may be elided, as in (15-b). This occurs in 265 (0.95%) of \( \text{pctb} \) trees.

(15) a. 我跟他说一句话

1st sg with 3rd sg say 1st sg mw speech

b. 我跟他说句話

1st sg with 3rd sg say mw speech

I will have a word with him.

This is a problem for the \( \text{ccg} \) analysis of the \( \text{Num} + \text{M} \) construction. Since we specify that the numeral “powers” the construction by selecting for a measure word (\( \text{M} \)) argument, no constituent can assemble the structure in (13) when the numeral is missing.

The two possibilities in \( \text{ccg} \) for the analysis of this construction are (a) categorial ambiguity on measure words, or (b) a unary rule \( \text{M} \to NP/\text{NP} \). We choose the unary rule, because this elision is possible regardless of the particular choice of measure word. Under the analysis through categorial ambiguity, an extracted corpus would only contain this analysis of those measure words which do occur with an elided numeral in the data, failing to predict its distribution.
He is an NBA player.

(16)  

\[
\begin{array}{c}
\text{self} \quad \text{is} \quad \text{个} \quad \text{NBA} \quad \text{球员} \\
\end{array}
\]

\[
\begin{array}{c}
NP \quad (S[\text{dcl}] \backslash \text{NP})/\text{NP} \\
M \quad \text{NP} / \text{NP} \\
\text{NP} \\
\text{NP} \\
S[dcl] \backslash \text{NP} \\
S[dcl]
\end{array}
\]

2.4.2 Bare versus non-bare nouns

Huang et al. (2008) gives the following examples to demonstrate that Chinese bare nouns can take on singular/plural, generic/definite/indefinite readings while requiring neither morphological marking nor obligatory syntactic marking such as articles.

(17)  

a. 狗 很 聪明  

dog very clever  

Dogs are clever.

b. 我 看到 狗  

1SG see-achieve:RES dog  

I saw a dog/dogs.

c. 狗 跑走 了  

dog run-away:RES SFP  

The dog/dogs ran away.

(all from Huang et al. (2008))

This state of affairs stands in contrast to English, in which a noun must undergo obligatory syntactic marking to obtain a definite reading:

(18)  

a. *Dog ran away.

b. The dog ran away.

In English CCGbank, the obligatory nature of determiners requires it to distinguish bare nouns \((N)\) from non-bare nouns \((NP[nb])\), with determiners being functors from bare to non-bare nouns \((NP[nb]/N)\), an analysis derived from Bierner (2001). A unary type-change rule \(N \rightarrow NP\) is also available to account for proper nouns, generic readings and other syntax which is determiner-less yet non-bare:
Chapter 2. The elements of Chinese CCG

(19) \[ \begin{array}{c}
  \text{dogs} \quad \text{ran away} \\
  \begin{array}{c}
    \text{N} \\
    \text{NP} \\
  \end{array} \\
  \begin{array}{c}
    \text{S[del]\NP} \\
    \text{S[del]} \\
  \end{array}
\end{array} \]

An analysis in the vein of English CCGbank which distinguishes bare and non-bare nouns has several advantages relating to over-generation. The English NP, once modified by a determiner, cannot undergo modification by further determiners:

(20)  
  a. the dog  
  b. *my the dog  
  c. *the any dog

An English CCGbank-style grammar predicts this by identifying the class of determiners with functors from bare to non-bare NP, blocking the modification of a non-bare NP by another determiner.

(21)  
  \[ \begin{array}{c}
    \text{*my} \\
    \text{the} \\
    \text{dog} \\
  \end{array} \]

Furthermore, distinguishing determiners from other noun modifiers such as adjectives blocks the modification of a non-bare NP by further noun modifiers:

(22)  
  \[ \begin{array}{c}
    \text{*red} \\
    \text{that} \\
    \text{dog} \\
  \end{array} \]

In contrast to English, Chinese does permit limited modification of the determiner. Huang et al. (2008) argues that in order to explain the grammaticality of (23-a), the structure of the D (determiner) head in Chinese must be more complex than in English, concluding that the D head must allow pre-modification by a proper noun. In (23-b), the determiner 那 that does not block further modification by 师范学校 model school, in contrast to the ungrammaticality of the English (22).

(23)  
  a. 我 喜欢 [张三 他 这个] 3SG like Zhangsang, this 
  b. 用 工 的 学生 diligent DE student
Adopting this analysis directly in the ccg setting would engender lexical ambiguity, since if the result of the Det + N unit already has category NP, further modifiers such as 共和党 Republican party would have to carry category NP/NP in addition to N/N.

Finally, the following pair indicates that definiteness is not obligatorily syntactically marked in Chinese:

\begin{align*}
\text{(24) a.} & \quad \text{“Dog ran away.} \\
\text{b.} & \quad \text{狗跑走} \\
& \quad \text{dog run-away:RES SFP} \\
& \quad \text{The dog ran away.}
\end{align*}

As such, we choose not to distinguish bare from non-bare NP anywhere in the ccg grammar. This results in the following categories:

\[
\langle \text{numeral} \rangle \vdash (NP/NP)/M \\
\langle \text{determiner} \rangle \vdash NP/NP
\]

Eliminating the bare/non-bare NP distinction may be undesirable for semantic interpretation, an objection also foreseen by Bierner (2001), who abandoned the use of distinct atoms N, NP in favour of an analysis through features $NP[+nb]$, $NP[-nb]$. The problem for semantic interpretation is that the interpretation of proper nouns (NP) is taken to be an entity ($e$ in Montagovian terms), while common nouns (N) are considered to be functions from entities to truth values ($\langle e, t \rangle$). Bierner (2001) addresses this concern by proposing that the semantics of all noun expressions, common and proper nouns alike, are semantically $\langle e, t \rangle$, and that proper nouns simply return true for their single referent and false otherwise. We adopt the same position.

In Chapter 7, we quantify the impact of this annotation decision in the parsing setting through parser-aided corpus conversion, and further justify our choice on practical efficiency grounds and the reduction in ambiguity through reducing the size of the lexicon.
2.4.3 Adjective-noun order, and adjectives versus stative verbs

English adjectives occur in two frames — noun modification, and predication.

(25) a. a massive tree  
    b. The tree is massive.

As in English, some Chinese adjectives are also capable of modifying nouns, giving them the syntactic type \( NP/NP \):

(26) a. a big tree  
    b. 大树  
       big tree

But unlike the distribution of adjectives in English, some Chinese adjectives form viable predicates on their own. In English, predication by an adjective requires the copula is:

(27) a. *He angry.  
    b. He is angry.  
    c. 他很生气  
       3SG very angry  
       He is very angry.

While the data suggest that such adjectives (the adjectival verbs of Li and Thompson (1989)) pattern after intransitive verbs, Huang et al. (2008) observes that selected adjectives can take an optional 对 dui complement (the semantic object of the adjective), while true verbs cannot:

(28) a. 这个工作对你很合适  
       this MW job dui 2SG very appropriate  
       This job is suitable for you.

    b. *这个工作很合适你  
       this MW job very appropriate 2SG  
       (intended) This job suits you.

Huang et al. (2008) uses this criterion to separate adjectives from verbs, arguing that they are separate lexical categories in Chinese. However, both adjectival predicates and intransitive verbs are both one-place predicates, and so receive the same category: \( S[dc]l\ NP \). We consider the applicability of the 对 dui complement to be a selectional restriction on partic-
2.4. Constructions headed by NP

<table>
<thead>
<tr>
<th>Type</th>
<th>Frequency</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purely VA</td>
<td>1213 (30.42%)</td>
<td>一样 same; 清楚 clear; 难 difficult; 高兴 happy; 够 enough; 足不 insufficient; 对 correct; 便宜 convenient</td>
</tr>
<tr>
<td>Purely JJ</td>
<td>2153 (53.99%)</td>
<td>总 total; 有关 relevant; 前 previous; 最后 last; 最高 highest; 大型 large-scale</td>
</tr>
<tr>
<td>Both frames</td>
<td>622 (15.60%)</td>
<td>大 big; 新 new; 高 high; 小 small; 好 good; 重要 important; 多 many</td>
</tr>
</tbody>
</table>

(a) Breakdown of adjectives in PCTB

<table>
<thead>
<tr>
<th>Type</th>
<th>Frequency</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purely VA</td>
<td>557 (7.12%)</td>
<td>里; afraid; loath; prone</td>
</tr>
<tr>
<td>Purely JJ</td>
<td>6358 (81.29%)</td>
<td>前; fiscal; third-quarter; chief</td>
</tr>
<tr>
<td>Both frames</td>
<td>906 (11.58%)</td>
<td>new; early; strong; first</td>
</tr>
</tbody>
</table>

(b) Breakdown of adjectives in PTB

Table 2.4: Partitioning adjectives into predicative, pure or mixed

ular adjectival predicates, rather than justifying distinct categories for adjectival predicates and for intransitive verbs.

The one-place predicate category $S[\text{dcl}]\NP$ is similar to the adjectival predicate category $S[\text{adj}]\NP$ in the English analysis of Hockenmaier (2003), except that the feature $\text{adj}$, necessary to prevent the over-generation in (27-a), is not necessary in Chinese due to the grammaticality of the corresponding example (27-c).

In both Chinese and English, some adjectives occur only as noun modifiers, some only as predicates, and some in both contexts. Those adjectives which are only capable of modifying nouns directly (e.g. 大型 large-scale) only carry the noun-modifying category $\NP/\NP$, those which can only serve as predicates (感谢 thankful) only carry the category $S[\text{dcl}]\NP$, and those which can fulfill both roles carry both categories.2

Table 2.4a partitions the lexical items tagged as adjectives (JJ) or stative verbs (VA) in PCTB into those occurring purely as JJ, purely as VA and those attested as both. The corresponding analysis in Table 2.4b, performed over PTB, shows that the distribution of the three classes is markedly different in English — only a small fraction of adjectives occur

---

2 As Duanmu (2000) notes, noun-modifying adjectives can yield unacceptability depending on the head (高山 tall mountain but not *高树 tall tree) while adjectival predicates are generally always acceptable (高耸, 树也高 the mountain is tall, the tree also). This argues for our analysis, in which the same lexical item may receive both categories, since different selectional restrictions can thus apply in each frame.
solely in a predicative frame, while the balance between purely predicative and purely noun-modifying adjectives is less skewed in Chinese.\(^3\)

However, in both languages, the set of adjectives which can function both as noun modifiers and as stative verbs are in the minority (15.60% in PCTB, 11.58% in PTB). Therefore, an analysis which distinguishes purely predicative adjectives from purely noun-modifying adjectives by category should not greatly increase ambiguity.

### 2.4.4 NP NP apposition

Chinese allows for two NPs to enter into an apposition relation, in which both apponends refer to the same entity (distinguishing it from coordination).

(29) ![apposition example](https://example.com/apposition1.png)

The two apponends are directly juxtaposed with no intervening punctuation, unlike English apposition, in which a comma or colon must generally intervene:

(30) a. 美国 总统 巴拉克·欧巴马

    America president Barack.Obama

b. The President of the United States Barack Obama

Because the annotation was absent from the Penn Treebank, Hockenmaier (2003) does not distinguish apposition from coordination in English CCGbank:\(^4\)

(31) ![apposition example](https://example.com/apposition2.png)

While the Penn Chinese Treebank does distinguish apposition structures, giving the first apponend the function tag -APP, the vast majority of apposition instances in PCTB, as in (29), do not separate the two apponends with punctuation, as the corresponding contexts

---

3 We note that these predicative-only English adjectives, such as rid, loath and prone, are those which take obligatory PP complements (rid of, loath to, prone to).

4 What we term NP NP apposition is referred to in Hockenmaier (2003) as the appositive noun phrase construction.
2.4. Constructions headed by NP

require in English. As such, we cannot import the English CCGbank analysis of apposition as coordination, since there is not necessarily a token separating the apponends.

To analyse NP NP apposition, we propose the following type-change rule:

**Rule 1 (NP-NP apposition type-change rule).**

\[
NP \text{ NP} \rightarrow NP
\]

2.4.5 \ S NP apposition

The juxtaposition of a non-finite S with an NP yields a construction which we call \ S-NP apposition, by analogy with the clausal apposition construction identified in English by Meyer (1992). The prevalence of \ S NP apposition is 1129 (4.03\%) of \ PCTB derivations.

\[
(32) \quad [\text{跨国\ 公司\ 在\ 山东\ 投资}]_S \text{ 势头}_NP
\]

multinational company in Shandong invest momentum

the momentum of multinational companies investing in Shandong

The fact that the result of \ S-NP apposition is semantically an NP suggests a first analysis through a type-change rule:

**Rule 2 (S-NP apposition type-change rule).**

\[
S \text{ NP} \rightarrow NP
\]

However, this rule would engender massive derivational ambiguity in a parser, as both input categories and the output category are extremely common. The rule is particularly harmful coupled with the rampant noun/verb ambiguity which we investigate later in Section 7.3 in the pos tagging setting.

A possible escape hatch is the requirement that the S in the apposition be non-finite. If we assume that all verbs underspecify for a “finiteness” feature [\pm\text{fin}], and the aspect markers (for example 了 \text{PERF}, 过 \text{EXP}) have the category of functors from non-finite to finite VPs, \ (S[+fin]\text{NP})/(S[−fin]\text{NP}), then the type-change rule can be constrained:

**Rule 3 (S-NP apposition type-change rule II).**

\[
S[−\text{fin}] \text{ NP} \rightarrow NP
\]

However, while a constraint on finiteness exists on the inputs to \ S-NP apposition, only 7289 (5.23\%) \ PCTB verbs receive any form of aspect marking. This is partly because overt
Table 2.5: Highest frequency head nouns in S-NP apposition in PCTB

aspect marking is blocked when the verb itself implies a bounded, completed activity (e.g. 寄 send), and also due to domain effects — newswire text reports on activities which are bounded and complete, and aspect marking is not in general obligatory. As such, the vast majority of PCTB verbs will be underspecified for finiteness, and Rule 3 will still be enormously productive.

A third analysis involves lexical ambiguity on the NP. Under this analysis, the NP receives a category of the form \((NP/\ast)\backslash S[dcl]\). However, because the head noun no longer receives its canonical category, modifiers of the head noun must receive non-canonical categories of the form \(((NP/\ast_1)\backslash S)/(NP/\ast_1)\backslash S)\), leading to greater sparsity. However, Table 2.5 suggests that the viability of the S-NP apposition construction is conditioned on the choice of head noun. In particular, the head noun must be abstract and must describe an activity (indicated by the non-finite S). Because the choice of head noun is not free, the impact of the lexical ambiguity engendered by this analysis is limited. Due to the massive derivational ambiguity involved in the analyses through type-change rules, we select the analysis through lexical ambiguity for Chinese CCGbank.

---

\[\text{---\textsuperscript{5}---}\]

The \(+\)-schematisation (§1.10) expands to a family of modifier categories. In this case, \(NP/\ast\) indicates the set \(\{NP/NP, (NP/NP)/(NP/NP), \ldots\}\). In common with Steedman (2000)’s \(\&\)-schematisation, the +-schematisation is understood not to yield an infinite set, but a finite set bounded by constraints on the depth of NP embedding. The maximum depth encountered in Chinese CCGbank is 5, encountered in 15 sentences; in English it is only 3 due to the scarcity of NP internal structure in the PTB.

\[\text{---\textsuperscript{6}---}\]

In English, all of the glossed head words in Table 2.5 can be followed by a PP headed by the semantically empty preposition of to yield semantics similar to the relationship between the S and NP in the Chinese construction.

\[\text{---\textsuperscript{7}---}\]

With the hat-CCG augmentation proposed by Honnibal (2010), we could ameliorate the categorial sparsity engendered by this analysis by giving the head noun the category \(NP^{NP}\backslash S[dcl]\).
Constructions involving the verb

The order-permuting crossed composition rules of ccg enable a simple and general analysis of the Chinese aspect particles, as well as the well-known separable V-O construction.

Aspect particles and other verbal modifiers

Tense and aspect are orthogonal attributes of finite verbs. Tense temporally pinpoints the action described by the verb relative to the moment of utterance, while aspect relates to the temporal context surrounding the action — for instance, whether the action was complete at the time it took place, whether it continued past that time, and so on (Comrie, 1976).

Most analyses of Chinese assert that it lacks morphological or grammaticalised tense marking, relying on context or temporal adjuncts to make tense distinctions (Comrie, 1976; Li and Thompson, 1989). But while Chinese tense marking is impoverished or absent, it does possess a rich system of aspect markers which appear as enclitics attached to verbs. The aspect markers attested in PCTB are given in Table 2.6. Prevalence is the proportion of sentences which contain an instance of each case.

Our analysis of aspect markers is analogous to the treatment in English CCGbank of post-verbal modifiers, assigning the category \((S\{dcl\}\parallel x(S\{NP\})\). However, consider that while this category suffices to modify the category \(S\{dcl\}\parallel NP\), backward application does not allow the aspect marker’s category to modify verbs with rightward arguments, including ditransitive verbs such as \(给\) give:

<table>
<thead>
<tr>
<th>Description</th>
<th>Frequency</th>
<th>Prevalence</th>
</tr>
</thead>
<tbody>
<tr>
<td>了 le</td>
<td>6215</td>
<td>5039 (18.01%)</td>
</tr>
<tr>
<td>着 zhe</td>
<td>875</td>
<td>778 (2.78%)</td>
</tr>
<tr>
<td>过 guo</td>
<td>283</td>
<td>271 (0.97%)</td>
</tr>
<tr>
<td>著 zhe</td>
<td>17</td>
<td>17 (0.06%)</td>
</tr>
</tbody>
</table>

Table 2.6: Chinese aspect markers attested in PCTB by frequency

Sybesma (2006) provides an argument against for Cantonese, while proposing that sentence final \(的 de\) in Mandarin can establish past tense in certain constrained environments.

著 zhe, the least frequent aspect marker in Table 2.6, is the 繁体字 (Traditional Chinese) equivalent of 着 zhe. While PCTB is standardised to Simplified Chinese character forms, a number of Traditional forms mistakenly remain in some sections reflecting non-Mainland Chinese media.
To generalise over verbs with any number and type of rightward arguments, we need the generalised composition rules of Steedman (2000), which allow for composition into functions of more than one argument such as the ditransitive verb category. The generalised backward crossed composition \( (< B^g_n) \) combinatory rules allow any number and type of rightward arguments to pass through the result category:

\[
(Y/Z)/S_1 X \rightarrow (X/Z)/S_1 \quad (< B^g_n)
\]

allowing aspect markers to compose into functors taking any number of rightward arguments:

\[
\begin{array}{c}
give \\
\rightarrow \\
\end{array}
\]

Hockenmaier (2003) notes that full context-sensitivity (corresponding to Type 1 grammars in the Chomsky hierarchy) is obtained by failing to bound the degree of generalised composition. In generalised composition of degree \( k \), the argument stack represented by the \( S \) in Rule 2.1 contains \( k \) arguments. As such, we license backward crossed composition to a degree equal to the maximum rightward valency of the language, which in Chinese is 3 (判处 sentence, which subcategorises for a patient, a prison sentence, and a duration). The maximum rightward valency in English CCGbank is also 3 (as in sold, which subcategorises for a theme, a recipient and a price).

However, the same facility cannot be extended to leftward verbal arguments — Chinese verbs may take subjects not only of category \( NP \) but also \( S[dcl] \), LCP, PP and QP:

\[
\begin{array}{c}
\text{给} \\
give \\
\rightarrow \\
\end{array}
\]

Hockenmaier (2003) notes that full context-sensitivity (corresponding to Type 1 grammars in the Chomsky hierarchy) is obtained by failing to bound the degree of generalised composition. In generalised composition of degree \( k \), the argument stack represented by the \( S \) in Rule 2.1 contains \( k \) arguments. As such, we license backward crossed composition to a degree equal to the maximum rightward valency of the language, which in Chinese is 3 (判处 sentence, which subcategorises for a patient, a prison sentence, and a duration). The maximum rightward valency in English CCGbank is also 3 (as in sold, which subcategorises for a theme, a recipient and a price).

However, the same facility cannot be extended to leftward verbal arguments — Chinese verbs may take subjects not only of category \( NP \) but also \( S[dcl] \), LCP, PP and QP:

\[
\begin{array}{c}
\text{给} \\
give \\
\rightarrow \\
\end{array}
\]
Because VP modifiers are distinguished from sentential modifiers in CCG by their form \((S\backslash X)/(S\backslash X)\) versus \(S/S\), their categories must mention a type of leftward argument. As such, while this analysis of verbal modifiers can generalise over any number of rightward arguments, they cannot generalise over the types of viable subject categories. Apart from proposing that the set of subject types available in Chinese are unifiable into a like type (the “predicative element” analysis of Carpenter (1992)), we are not aware of an analysis of post-verbal modifiers which generalises over all verb frames. As such, we rely on lexical ambiguity, assigning modifier categories \((S\backslash X)/(S\backslash X)\) for each \(X\) which is a valid subject type.

## 2.5.2 V-O forms

A class of Chinese verbs (the V-O forms of Chao (1968) and Packard (2000), and the cognate object verbs of Ross (1998)) are words diachronically derived from \(V+N\) character combinations (e.g. 吃饭 eat-rice) which exhibit both word-like (36-a) and phrase-like (36-b) properties.

(36)  

a. 今天 晚上 在 家 吃饭 吗  
today night at home eat-rice q  
Are (you) having dinner at home tonight?  

b. 你们 饭 吃 了 吗  
1PL rice eat ASP q  
Dinner, have you had (it) yet?

Since various kinds of material (aspect particles, numerals and other noun modifiers) can intervene between the \(V\) and \(N\), this phenomenon is also known as ionisation or separability (离合).

While researchers have disagreed on the diagnostics for separating V-O verbs from \(V+N\) phrasal combinations, a characteristic common to several analyses is that the meaning of a V-O verb is non-compositional, in contrast to the \(V+N\) phrasal combination (Chao, 1968; Li and Thompson, 1989; Packard, 2000). Thus, 吃饭 eat-rice has the non-
compositional meaning have dinner as a V-O verb, and eat a meal of rice as a V + N phrasal combination.

Like V + N phrasal combinations, however, V-O verbs are separable, in the sense that 1) aspect particles such as 过 DUR and 了 ASP can modify the V part directly, and 2) the O can receive noun modifiers, such as an indefinite numeral phrase.

(37)  

a. 在屋 里 吃 了 饭
    had dinner at home
    at house inside:LC have- ASP -dinner

b. 你们 大家 抬 抬 手 , 原 个 谅 吧。
   1PL everybody lift-lift hand , for- MW -give sfp .
   Let’s all lift up our hands and show a little forgiveness, okay?
   (from Fang (1994), quoting 《新儿女英雄传》 by 袁静、孔厥)

While it may seem natural to analyse a V-O form like 饭 rice as the syntactic direct object of a transitive 吃 eat in both contexts, only when the O is interpreted as a meal of rice is this true:

(38)  

你 吃 了 什么? 我 吃 了 饭。

What did you eat?
   \{ I had rice.
   \{ "I had dinner.

This is especially obvious when the O is no longer a free morpheme in the modern language (觉 in 睡觉 sleep), or indeed a phonetic loan re-analysed as a V-O structure, as 幽默 humour in (39-b) is.

(39)  

a. 睡 了 三 个 钟头的 觉
   sleep ASP three MW hour DE sleep
   slept for three hours

b. 我 幽 了 他 一 默
   I hu- ASP 3SG one -mour
   I teased him a bit
   (example from Packard (2000))

We present an analysis which allows our ccg analysis to distinguish V-O word structure and V + N phrasal combination, preserving the separability property of the V-O word
construction, by introducing a new atom $O$ which represents the $O$ element of the V-O construction.

This allows the separated $O$ to participate in topicalisation (41-a), and for aspect particles to intervene between the $V$ and $O$ as in (41-b), at the cost of introducing modifier proliferation on modifiers of $O$:

Under hat-ccg, the $O$ of a V-O form would receive a category $N^O$ to allow the $O$ to receive noun modifiers while requiring that it eventually become the complement of a $V$.

2.6 Coordination phenomena

ccg accounts of coordination phenomena such as non-constituent and argument cluster coordination are hallmarks of the formalism. We discuss the extent to which these ana-
Chapter 2. The elements of Chinese ccg

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Prevalence</th>
<th>Can coordinate</th>
</tr>
</thead>
<tbody>
<tr>
<td>和 he</td>
<td>5065</td>
<td>4159 (14.86%)</td>
</tr>
<tr>
<td>与 yu</td>
<td>1476</td>
<td>1311 (4.68%)</td>
</tr>
<tr>
<td>及 ji</td>
<td>806</td>
<td>726 (2.59%)</td>
</tr>
<tr>
<td>以及 yiji</td>
<td>546</td>
<td>535 (1.91%)</td>
</tr>
<tr>
<td>并 bing</td>
<td>532</td>
<td>522 (1.87%)</td>
</tr>
<tr>
<td>或 huo</td>
<td>457</td>
<td>422 (1.51%)</td>
</tr>
<tr>
<td>至 zhi</td>
<td>270</td>
<td>229 (0.82%)</td>
</tr>
<tr>
<td>而 er</td>
<td>267</td>
<td>261 (0.93%)</td>
</tr>
<tr>
<td>到 dao</td>
<td>170</td>
<td>161 (0.58%)</td>
</tr>
<tr>
<td>并且 bingqie</td>
<td>149</td>
<td>146 (0.52%)</td>
</tr>
<tr>
<td>又 you</td>
<td>79</td>
<td>68 (0.24%)</td>
</tr>
<tr>
<td>也 ye</td>
<td>74</td>
<td>75 (0.27%)</td>
</tr>
<tr>
<td>跟 gen</td>
<td>74</td>
<td>73 (0.26%)</td>
</tr>
</tbody>
</table>

Table 2.7: Chinese coordination words attested in PCTB by frequency

yses hold in Chinese, and consider differences between English and Chinese coordination which affect our choice of CCG analysis.

2.6.1 The syntax of coordination

The Coordinate Structure Constraint (csc) states that coordination is possible between constituents of like type (Ross, 1967; Haspelmath, 2004). The CCG reflex of the csc states that coordination is possible between spans of like category (Steedman, 2000). We adopt the CCG coordination analysis from English CCGbank (Hockenmaier, 2003), a right-branching binarised version of the ternary coordination schema originally proposed by Steedman (2000).10

Rule 4 (Right-branching binarised coordination).

\[
X X[\text{conj}] \rightarrow X \\
\text{conj } X \rightarrow X[\text{conj}]
\]

---

10 Zhang (2010) argues on syntactic and prosodic grounds that 1) coordination cross-linguistically results in binarised rather than k-way branching structures, and that 2) languages like Japanese generate left-branching coordination structures. A left-branching binarised CCG coordination schema could exist in a language like Japanese.
Lexicalising coordination

Several CCG analyses, including the analysis embodied by Italian CCGbank (Bos et al., 2009) and the analysis of Toba Batak by Baldridge (2002) rely on an analysis of coordination which assigns the coordination word a category of the form $(X_1 \backslash X_2) / X$, instead of relying on extra-combinatory rules such as Rule 4.

One benefit of lexicalising coordination, apart from the minimisation of rules not predicted by the theoretical base, is that distinctions between various coordination words can be expressed in the lexicon. For instance, while the English coordination word and is flexible enough to coordinate conjuncts of any type, languages like Japanese call on different coordination words, and indeed may use different strategies altogether to coordinate different types of conjuncts (Zhang, 2010). For instance, Japanese has a coordination word し shi which conjoins clauses, a coordination word と to which only conjoins NPs, and a distinct verb form for conjoining VPs:

(42)  a. 宿題 も して いないし、学校の準備 も して homework also do:GER be:NEG and, school gen preparation also do:GER いない。
     be:NEG (I) haven't done (my) homework, and (I) haven't gotten ready for school either.

b. 今日 は カレーライス と ハンバーグ です。
   today TOP curry.rice and minced.beef is
   Today, it’s curry rice and minced beef.

c. 本 を 読んだり 音楽 を 聞いたり した。
   book ACC read:CONJ music ACC listen:CONJ do:pst
   (I) read a book and listened to music.

Similarly in Chinese, some coordination words, such as 并 bing, only conjoin VP:

(43)  在 世界上 率先 研究 成功， 并 具有 国际 先进 水平
take the initiative achieving success globally through research, and possess a international standard of advancement

take at world on initiative research succeed, and have international advanced standard

The high-frequency coordinators attested in PCTB, together with their frequency and the types they may conjoin, are given in Table 2.7.
This is a marked departure from English, whose high-frequency coordinators (and, or, as well as, rather than) are essentially unconstrained in terms of the syntactic type of what they may conjoin. Under a lexicalised coordination analysis, the categories assigned to each coordination word encode the types of the conjuncts which that word can coordinate, allowing fine-grained distinctions to be made in the lexicon.\(^{11}\)

Chinese is known to have two disjunctive coordination words, 还是 haishi and 或者 huo\(z\)he. A disjunction separated by 或者 huo\(z\)he is interpreted as logical disjunction (\(\text{vel}\)). On the other hand, a disjunction separated by 还是 haishi can only be interpreted as a question asking which of the conjuncts the proposition is true for (44-c), or else as the complement of a regardless-clause (44-e) (Li and Thompson, 1989; Zhang, 2010).

\[\]
\[
\]

(44) a. 你 喜欢 绿色 或者 蓝色 吗? 喜欢。
2\(sg\) like green huo\(z\)he blue q? like.
Do you like green or blue? Yes.

b. 我 喜欢 绿色 或者 蓝色。
1\(sg\) like green huo\(z\)he blue.
I like green or blue.

c. 你 喜欢 绿色 还是 蓝色? 喜欢 蓝色。
2\(sg\) like green haishi blue? like blue.
Do you like green, or blue? I like blue.

d. *我 喜欢 绿色 还是 蓝色。
1\(sg\) like green haishi blue.
(intended) I like green, or blue.

e. 无论 贫穷 还是 富有, 健康 还是 疾病
no matter poor haishi rich , healthy haishi sick
whether poor or rich, in good health or in bad

还是 haishi forces a question reading of the sentence — only the question reading (44-c) is possible, and not the declarative reading (44-d).

We propose that the coordinator 还是 haishi imbu\(e\)es its conjuncts with a feature \([\text{whc}]\) (\(\text{which conjunct}\)) which particular verbs must subcategorise for. Such verbs would carry the category \((S[q]\backslash NP)/NP[\text{whc}]\). For instance, the verb 喜欢 like in the example below would subcategorise for a which-conjunct NP, represented as a category \(NP[\text{whc}]\) which is derivable from coordination with 还是 haishi.

\[^{11}\text{Indeed, selectional restrictions on Chinese coordinators extend beyond the type of conjuncts they may coordinate; the coordinator 兼 jian can only coordinate two predicates referring to the same real-world referent. See Huang et al. (2008) for more data and an analysis.}\]
2.6. Coordination phenomena

(45)  
NP  (S[q]\NP)/NP[whc]  NP  (NP[whc]\NP)/NP  NP
    NP[q]\NP  NP[whc]
    S[q]\NP  S[q]

Thus, words introducing a regardless-clause such as 无论 regardless can subcategorise specifically for NP[whc] complements, with the category (S/S)/NP[whc].

While the extra-combinatory coordination rules of Hockenmaier (2003) may be sufficient for English on account of the permissiveness of the coordination word and, languages such as Chinese, in which coordination words select for different types of conjuncts, and in which coordination words differ semantically, benefit from a lexicalised analysis of coordination.

2.6.2 List commas

Chinese distinguishes two types of commas — one used exclusively to offset the conjuncts in a coordination structure (， —顿号 dunhao), and another (， —逗号 douhao) for all other uses (the indication of prosodic breaks, and disambiguation).12

(46)  a. 再则 ，服务 快捷 、 灵活 。
    Besides, service quick LCM nimble
    Besides, the service is quick and agile.

    b. 谈到 感情 、 婚姻 、 就 复杂 了 。
    When it came to talking about feelings and marriage, things got complicated.

While this suggests that we could restrict coordination to conjuncts joined by (the 顿号 dunhao), this is not possible in general — loose coordination between IP or between VP is not considered enumeration, and so the clause-separating commas discussed later in Section 2.8.1 are realised as (逗号 douhao) and not (顿号 dunhao). Nevertheless, in the same way that a lexicalised analysis of coordination is able to restrict particular coordinators to coordinating only particular types of conjuncts, such an analysis could restrict

12 In this work, we gloss the 顿号 dunhao token 、 as LCM (list comma) to emphasise its role as a list delimiter.
(逗号 douhao) to coordinating VP or IP conjuncts only, reducing derivational ambiguity.

2.6.3 Right node raising

Right node raising (RNR) is the sharing of a rightward argument among more than one functor, and occurs in 510 (1.82%) of PCTB trees. In (47-a), the single argument the export of valuable timber is shared between two verbs: restrict and control.

(47) a. 草案也提出国家要禁止限制出口珍贵木材
draft also present nation need ban LCM control export valuable timber
The draft also points out that the nation must restrict and control the export of valuable timber.

b. 武装森林警察部队执行预防和扑救森林火灾的任务
armed forest police squad execute protect and rescue forest fire DE task
The armed forest police squad carries out the task of protecting and defending from forest fires.

Right node raising is constrained by the across-the-board condition (Ross, 1967), which stipulates that “the element adjoined to the coordinate node must occur in each conjunct”. In CCG, this is a corollary of the fact that coordination operates on units of like category (Steedman, 2000). Because constituents with the same category seek the same arguments, the fact that each conjunct has the same category ensures that each conjunct is “missing” an element of like type.

The CCG analysis of the RNR portion of (47-b) is thus:

\[
(S[dcl]\, NP)/NP \succ NP/NP \succ NP/NP \succ S[dcl]\, NP \succ NP/NP \succ NP
\]
2.6.4 Argument cluster coordination

The ccg analysis of argument cluster coordination in English is a hallmark of the formalism. The B-T combinatory calculus allows the assignment of principled types to non-traditional constituents such as the clusters arising from argument cluster coordination:

(49)  
```
I made Egbert cake and Hortense a salad
```

However, it is unclear if cases of argument cluster coordination corresponding to the above English example are grammatical in Chinese. An informal poll of five native Mandarin Chinese speakers living in Australia and China suggests that while (50-a) remains interpretable, the coordination must be between VPs for the sentence to be grammatical (50-b).

(50)  a. 张三给李四一个香蕉，王五一个苹果
     Z. give ASP L. one MW banana LCM W. one MW apple
     ??Zhangsan gave Lisi a banana, and Wangwu an apple.

     b. 张三给了李四一个香蕉，给了王五一个苹果
     Z. give ASP L. one MW banana LCM give ASP W. one MW apple
     Zhangsan gave Lisi a banana, and Wangwu an apple.

However, a construction similar to argument cluster coordination, which pairs NPs and quantities of that NP, is attested in the Penn Chinese Treebank:

(51)  开发油田三百五十个，气田一百十
     develop oilfield 350 MW, gas field 110 MW
     to develop 350 oilfields and 110 gas fields

In the HPSG conversion of the Penn Chinese Treebank by Yu et al. (2010), this argument cluster coordination-like construction is cited as one which is difficult for current formu-
lations of HPSG to account for. We use the argument cluster coordination analysis used by Hockenmaier (2003) for English CCGbank, which yields:

\[
\text{开发} \quad \text{油田} \quad 350 \text{ MW} \quad , \quad \text{气田} \quad 110 \text{ MW}
\]

where we abbreviate \((S[\text{def}]\backslash NP)/QP \equiv QTV\) and \(QTV/NP \equiv QDTV\) for brevity. Because the two conjuncts have the same category \(VP\backslash QDTV \equiv VP\backslash QTV\backslash (VP/QP)/NP\), they may be coordinated. The argument cluster then consumes the ditransitive verb \(QDTV \equiv (VP/QP)/NP\), saturating its two arguments.

We assume that the restriction which makes (50-a) marginal or questionable is a constraint on the form of possible type-raised categories, a facility also invoked by Steedman (2000) to constrain the types of possible non-constituent conjuncts.

### 2.6.5 Unlike coordination phrases

Unlike coordination is coordination between conjuncts of syntactically different types: for example, \(NP\) and \(ADJP\) in (53-a), and \(NP\) and \(IP\) in (53-b).13

\[
\text{a. Pat is a Republican and proud of it.}
\]
\[
\text{b. 一 支 如此 高 质素 、 纪律 严明 的 驻军}
\]
\[
\text{one MW such high quality discipline strict garrison}
\]
\[
\text{a garrison of such high quality and strict discipline}
\]

In the PCTB annotation, if the conjuncts in a coordination are not all of like type, the node which dominates all the conjuncts receives the phrase-level tag \(UCP\). \(UCP\) is present in 438 (1.57%) of PCTB trees.

The most common UCP type in PCTB is coordination between \(NP\) and \(IP\) conjuncts, where the \(IP\) takes on a nominalised reading:

---

13 The translation for (53-b) suggests like-type coordination, but here \(严明\) strict is a stative verb taking a subject \(纪律\) discipline, forming a sentence.
Because coordination has widely been defined across formalisms as an operation on elements of *like type* (Steedman, 2000; Haspelmath, 2004), the analysis of UCP is problematic. One solution is that the “unlike types” of UCP are actually unifiable into like types (Prolo, 2006; Komagata, 2002). The solution of Komagata (2002) suggests that all types which can occur together with predicative be — in English, the weak NPs of Milsark (1977), the predicative adjectives of Quirk et al. (1985), and some PP — share the ccg category *PE* (predicative element). Since these are precisely the UCP types in English, this reduces UCP to plain coordination between *PE*s.

While the *PE* solution is theoretically pure, the implications for categorial ambiguity, as well as over-generation, are considerable: under the scheme, all types licensed for UCP collapse to *PE*, and there is no way to distinguish, for instance, a PP modifier from an NP modifier, as both would receive the category *PE/PE*, licensing the ungrammatical:

\[
\begin{align*}
\text{red} & \quad \text{in} \quad \text{the} \quad \text{car} \\
\text{PE/PE} & \quad \text{PE/NP} \quad \text{NP/PE} \quad \text{PE} \\
\text{NP} & \quad \text{PE} \\
\text{PE} & \quad \text{PE}
\end{align*}
\]

Another possibility is to consider that *利用 外资* *use foreign-investment* has the *form* of a verbal predicate, but the *function* of an NP. The hat-ccg of Honnibal (2010) lets us lexically specify unary type-changes from each conjunct, until all conjuncts have the same (functional) type:
Chapter 2. The elements of Chinese ccg

(56)  China’s economy and the use of foreign investment

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>conj</th>
<th>(S[decl]\NP)\NP</th>
<th>NP</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N/N$</td>
<td>$N$</td>
<td>$\Rightarrow$</td>
<td>$S[decl]\NP$</td>
<td>$\Rightarrow$</td>
</tr>
<tr>
<td>$NP$</td>
<td>$NP$</td>
<td>$\Rightarrow$</td>
<td>$NP[conj]$</td>
<td></td>
</tr>
</tbody>
</table>

In the absence of hat-ccg, relaxing the same-category restriction of regular coordination yields a CCGbank-style analysis through binary type-change rules:14

Rule 5 (Unlike coordination rule schema).

$$conj Y \rightarrow X[conj], \text{ for } X \text{ and } Y \text{ licensed for UCP}$$

The right hand side of the rule must assign a set head to represent the combination of the heads of $X$ and $Y$; in UCP the set head is the union of the heads of $X$ and $Y$.

We select the analysis through binary type-change rules for Chinese CCGbank.

2.7 Sentence-level phenomena

We consider the ccg analysis of yes-no (polar) questions, and the analysis of direct quoted speech, a very common convention in the text types which PCTB represents.

2.7.1 Yes-no questions

The strategy for forming a yes-no question from a declarative sentence in Chinese is to append the sentence-final question marker $\text{吗} ma$.

English polar questions undergo verb-subject inversion, allowing the auxiliary verb to effect the inverted order and cause the sentence to gain the polar question category $S[q]$:15

---

14 The set of categories licensed for UCP in Chinese CCGbank is given in Appendix B. We selected these by adding all conjunct types annotated as UCP in the PCTB down to cutoff $5$.

15 Questions which do not cause verb-subject inversion (e.g. This is Japan?) are analysed in English CCG-bank by treating the question mark token as absorbed punctuation (see §1.8.2).
English CCGbank prefers this analysis over functional punctuation (\( \rightsquigarrow S[q] \backslash S[\text{inv}] \)) because particular subordinators take embedded polar questions as complements:

(58) Chinese polar questions do not effect word order changes, [ nor ](S[S]/S[q]) [ do they occur as embedded phrases ]S[q].

Clark et al. (2004) also presented CCG analyses of wh-word questions in which the categories of wh-words trigger word order changes, rather than relying on punctuation to indicate question syntax.

Chinese polar questions do not effect word order changes, nor do they occur as embedded phrases, so we analyse the particle 问 ma as a functor from a declarative sentence to a polar question:

(59) Are the children tired?

孩子们 累 问 ？

children tired Q ？

NP S[dcl] \backslash NP S[q] \backslash S[dcl] 

S[dcl] 

S[q] 

Polar questions are rare in PCTB, occurring in 161 (0.58\%) of PCTB trees.

2.7.2 Direct quoted speech

In common English newswire style (“...,” he said or “...,” said John) the quoted span (the object of reporting verb said) is situated at sentence-initial position. In light of its high frequency, the analysis used in English CCGbank assigns a leftward-looking category for reporting verbs such as said which take as arguments the speaker and the quote:
Chapter 2. The elements of Chinese CCG

The apple juice scene was so funny; John said:

\[
\frac{\text{S[decl]}}{\text{NP (S[decl] \text{\&} S[decl]) \text{\&} NP}} \frac{\text{S[decl] \text{\&} S[decl]}}{\text{S[decl] \text{\&} S[decl]}} \frac{\text{S[decl] \text{\&} S[decl]}}{\text{S[decl] \text{\&} S[decl]}}
\]

While this fronted style is rarer in Chinese newswire, it is nevertheless available. However, we diverge from English CCGbank by choosing an analysis for fronted direct quoted speech which allows the reporting verb to retain its canonical category. This analysis, which treats the fronted constituent as the result of topicalisation, has the advantage of being the CCG analogue of the analysis of fronted direct quoted speech used in the Penn Chinese Treebank (Xue et al., 2000).

As such, Chinese CCGbank analyses fronted direct quoted speech by licensing \text{S[decl]} as a candidate for the extra-combinatory topicalisation rule. This enables the following analysis through the rule instantiation \text{S[decl]} \rightarrow \text{S/(S/S[decl])}:

\[
\frac{\text{S[decl]}}{\text{S/(S/S)}} \frac{\text{S[decl]}}{\text{S/(S/\text{NP})}} \frac{\text{S[decl]}}{\text{S/(S[decl])}}
\]

The annotation guidelines in Xue et al. (2000) also provide an analysis for discontinuous quoted speech, such as:

\[
\text{「 总之不是台湾, 」} \quad \text{"In any case it's not Taiwan,"}
\]

A CCG analysis through scrambling can be developed (Hoffman, 1996):

\[
\text{我 相信 ,} \quad \text{张三 说,} \quad \text{你们 会 成功} \quad \text{I believe,} \quad \text{Z. said,} \quad \text{2PL will succeed}
\]

"I believe," Zhangsan said, "that you will succeed."

A CCG analysis through scrambling can be developed (Hoffman, 1996):

\[
\frac{\text{S[decl]} \text{\&} \text{believe} / \text{S[decl]}}{\text{S[decl]} \text{\&} \text{say} / \text{S[decl]}} \frac{\text{S[decl]} \text{\&} \text{succeed}}{\text{S[decl]} \text{\&} \text{succeed}}
\]

\[
\frac{\text{S[decl]} \text{\&} \text{believe/said} / \text{S[decl]}}{\text{S[decl]} \text{\&} \text{succeed}}
\]

A CCG analysis through scrambling can be developed (Hoffman, 1996):

\[
\frac{\text{S[decl]} \text{\&} \text{believe} / \text{S[decl]}}{\text{S[decl]} \text{\&} \text{say} / \text{S[decl]}} \frac{\text{S[decl]} \text{\&} \text{succeed}}{\text{S[decl]} \text{\&} \text{succeed}}
\]

\[
\frac{\text{S[decl]} \text{\&} \text{believe/said} / \text{S[decl]}}{\text{S[decl]} \text{\&} \text{succeed}}
\]

A CCG analysis through scrambling can be developed (Hoffman, 1996):

\[
\frac{\text{S[decl]} \text{\&} \text{believe} / \text{S[decl]}}{\text{S[decl]} \text{\&} \text{say} / \text{S[decl]}} \frac{\text{S[decl]} \text{\&} \text{succeed}}{\text{S[decl]} \text{\&} \text{succeed}}
\]

\[
\frac{\text{S[decl]} \text{\&} \text{believe/said} / \text{S[decl]}}{\text{S[decl]} \text{\&} \text{succeed}}
\]
However, we do not analyse the discontinuous quoted speech construction, since only one instance exists in the Penn Chinese Treebank.

### 2.8 Punctuation

For the analysis of punctuation which does not project dependencies, such as the sentence-final period, English CCGbank uses an absorption analysis which attaches the punctuation at any level in the derivation, leaving the modified category unchanged:

**Rule 6** (Punctuation absorption rules). For any absorption punctuation category α, and any category X:

\[
\alpha X \rightarrow X \\
X \alpha \rightarrow X
\]

This is consistent with the analysis in English CCGbank (Hockenmaier, 2003). We propose that the tokens in Table 2.8 are candidate absorption punctuation categories in Chinese.

<table>
<thead>
<tr>
<th>Freq.</th>
<th>Token</th>
<th>Description</th>
<th>Freq.</th>
<th>Token</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>51525</td>
<td>，</td>
<td>逗号 douhao</td>
<td>1275</td>
<td>»</td>
<td>Close double chevron</td>
</tr>
<tr>
<td>23559</td>
<td>。</td>
<td>Period</td>
<td>1274</td>
<td>«</td>
<td>Open double chevron</td>
</tr>
<tr>
<td>3546</td>
<td>「</td>
<td>Open square quote</td>
<td>1038</td>
<td>；</td>
<td>Semicolon</td>
</tr>
<tr>
<td>3542</td>
<td>」</td>
<td>Close square quote</td>
<td>847</td>
<td>？</td>
<td>Question mark</td>
</tr>
<tr>
<td>2148</td>
<td>“</td>
<td>Open double quote</td>
<td>308</td>
<td>!</td>
<td>Exclamation mark</td>
</tr>
<tr>
<td>2134</td>
<td>”</td>
<td>Close double quote</td>
<td>132</td>
<td>「</td>
<td>Open double square quote</td>
</tr>
<tr>
<td>2083</td>
<td>:</td>
<td>Colon</td>
<td>132</td>
<td>「</td>
<td>Close double square quote</td>
</tr>
<tr>
<td>2062</td>
<td>（</td>
<td>Open round bracket</td>
<td>83</td>
<td>’</td>
<td>Open single quote</td>
</tr>
<tr>
<td>2051</td>
<td>）</td>
<td>Close round bracket</td>
<td>80</td>
<td>‘</td>
<td>Close single quote</td>
</tr>
</tbody>
</table>

Table 2.8: Chinese CCGbank candidate absorption categories

---

16 Italian CCGbank (Bos et al., 2009), however, intentionally eschews extra-combinatory rules of absorption by giving intra-sentential punctuation categories of the form X\X (where X is the category of the unit that the punctuation is right-joined to), and sentence-final periods a special category T\S which blocks further modification (T being a top-level category). For Chinese CCGbank, we retain the absorption analysis for compatibility with parsers and tools developed for English CCGbank.
2.8.1 Clause-separating commas

The phenomenon in which full sentences are separated by commas is referred to as *comma splice*. While prescriptively condemned by some English style guides, comma splice sentences are particularly prevalent in Chinese writing.

(64)  
- I went to the market, the bananas were expensive.
- 我去超市 买东西，香蕉特别贵
  1sg go supermarket buy thing，banana special expensive
  I went to the supermarket, the bananas were especially expensive.

In our analysis, comma splices are treated as coordination between two full sentences, requiring no special analysis.

2.8.2 Sentence modifiers of sentences

The juxtaposition of two full sentences in Chinese implies a cause and effect relationship between the first and the second, which may be contrafactual (Fang, 1994).

(65)  
- 没有和平环境，任何建设事业都无从谈起。
  not-have peace environment，any construction project all unable talk-up.
  Without an outlook of peace, there can be no discussion of any construction projects.

When the two sentences are delimited by a comma, we can analyse the comma as a functor which introduces the sentential modifier:

(66)  
\[
\begin{array}{c}
\text{没有和平环境} \\
S[dl] \\
\hline
\text{任何事业都无从谈起} \\
(S/S)\backslash S[dl] \\
\hline
S[dl]
\end{array}
\]

This analysis is reminiscent of the punctuation-cued type-change rules in English CCG-bank for the analysis of extraposed NP modifiers (Hockenmaier, 2003):

(67)  
\[
\begin{array}{c}
\text{No dummies} , \\
NP \\
\hline
S/S \\
\hline
S[dl]
\end{array}
\]

, the drivers pointed out they still had space.
2.9. Verb compounding strategies

which is analogous to the functor analysis we propose, except that we choose lexical ambiguity rather than derivational ambiguity.

2.8.3 Parentheticals

A parenthetical expression is a phrasal category delimited by paired punctuation tokens enveloping the phrase, or otherwise delimited by a punctuation token at the phrase’s left boundary. Parenthetical expressions occur right-adjoined to another phrase.

following Hockenmaier (2003), we interpret parentheticals as being headed by the opening punctuation token, with the right punctuation token (if present) absorbed:

2.9 Verb compounding strategies

The Penn Chinese Treebank annotation distinguishes the six verb compounding and derivation strategies in Table 2.9.

2.9.1 Verb potential construction (VPT)

In this construction, a polysyllabic verb receives an infix 得 de or 不 bu, producing a verbal compound meaning can V or cannot V respectively.

He couldn't even sort out how old the three kids were, and what year they were in.
<table>
<thead>
<tr>
<th>Tag</th>
<th>Example</th>
<th>Prevalence</th>
</tr>
</thead>
<tbody>
<tr>
<td>VRD</td>
<td>煮熟 cook done</td>
<td>2025 (7.24%)</td>
</tr>
<tr>
<td>VCD</td>
<td>投资设厂 invest &amp; build-factory</td>
<td>1007 (3.60%)</td>
</tr>
<tr>
<td>VSB</td>
<td>规划建设 plan then build</td>
<td>1005 (3.59%)</td>
</tr>
<tr>
<td>VPT</td>
<td>离得开 leave able away</td>
<td>384 (1.37%)</td>
</tr>
<tr>
<td>VNV</td>
<td>去不去 go or not go</td>
<td>188 (0.67%)</td>
</tr>
<tr>
<td>VCP</td>
<td>确认为 confirm as</td>
<td>148 (0.53%)</td>
</tr>
</tbody>
</table>

Table 2.9: Verb compounding strategies in the Penn Chinese Treebank by frequency in PCTB

b. 我 不 怕 菜 卖 不 出 去 , 就 怕 它 长 不 出 来
   1SG not scared vegetables sell BU out:RES, but scared 3SG grow BU out:RES
   I'm not afraid that my vegetables won't sell — I'm afraid that they won't grow.

c. 没有 任何 站 得 住 脚 的 理由
   not-have any stand DE still foot DE reason
   not have any reason to stand still

Three possibilities for the CCG analysis are:

- to have the infix particle 得 de or 不 bu collect the verbal arguments on either side
- a morphologically decomposed analysis where the infixation is represented linearly
- to merge the internal structure of the VNV construction into an atomic lexical item

The first option, proposing that the combination VNV is headed by the infix N (that is, the particle 得 de or 不 bu), calls for no extra-combinatory rules, and preserves the tokenisation used in PCTB. However, because the verb root is cleaved into two tokens, the lexicon becomes sparser, as each cleaved part forms a lexical item independent from the original verb root.

\[
\text{be unable to complete}
\]

\[
\begin{array}{c}
\text{作}\\
\text{do}
\end{array}
\begin{array}{c}
\text{不}\\
\text{BU}
\end{array}
\begin{array}{c}
\text{了}\\
\text{complete:RES}
\end{array}
\]

\[
\frac{TV}{(TV\backslash TV)/TV} \quad \frac{TV}{TV} \quad TV
\]

\[
\frac{TV}{TV} \quad TV
\]

\[
TV
\]
The morphologically decomposed analysis proposes that the syntactic component sees the linear sequence of tokens \( V_1 V_2 «BU» \), the spell-out of which is the infixed representation \( V_1 «BU» V_2 \). This analysis is still consistent with the Principle of Adjacency of Steedman (2000), which only requires that combinatory rules apply to string-adjacent entities, at the cost of positing an additional level of representation. However, this preserves the unity of the verb root, because infixation is posited to apply after syntactic analysis.

The last option, merging the internal structure into an atomic lexical item, treats VNV as a morphological operation invisible to syntax.\(^{17}\) We adopt this last position, which avoids the lexical ambiguity of the cleaving analysis, and the additional level of representation required for the morphologically decomposed analysis. We note that the appropriate contextual predicates in a tagger should easily be able to distinguish cases of VNV from lexical verbs and instances of other verb compounding strategies.

### 2.9.2 The V-bu-V construction (VNV)

The V-bu-V (VNV) construction forms a question from a declarative sentence by applying the infix 不 \( bu \) to its main verb. Diachronically, the VNV construction is a lexicalisation of a disjunctive question construction:

\[
\begin{align*}
(72) & \quad \text{你 喜欢(是)不 喜欢 吃 面条} \\
& \quad \text{25G like (or) not like eat noodles} \\
& \quad \text{a. you like (or) not like eat noodles} \\
& \quad \text{b. you like not like eat noodles} \\
\end{align*}
\]

Disyllabic verbs \( V_1 V_2 \) participate in the construction by being cleft and copied: \( V_1\)-bu-\( V_1 V_2 \). While both constructions involve infixation, the V-bu-V construction is distinguished from the VPT construction above in two ways: 1) V-bu-V heads a polar question, while VPT still heads a declarative sentence, and 2) V-bu-V involves copying, while the \( V_2 \) element in VPT is a resultative complement.

However, we argue that merging the internal structure of V-bu-V into an atomic lexical item is the most appropriate analysis in CCG, following the same argument as in the previous section. Under this analysis, the merged V-bu-V lexical item should carry a polar question feature:

\[
\begin{align*}
\text{喜欢 like} & \quad \vdash (S[dcl]\backslash NP)/S[dcl] \\
\text{喜不喜欢 like:VNV} & \quad \vdash (S[q]\backslash NP)/S[dcl]
\end{align*}
\]

\(^{17}\) This re-analysis was first suggested to us by Mark Steedman (p.c.).
2.9.3 Verb resultative complements (VRD)

Verb-resultative compounds, composed of a verb together with a resultative/directional complement, give the resulting state or position of the activity of the verb (Li and Thompson, 1989).

(73) a. 建设 成
build -become:res
build into
b. 保存 完整
preserve -intact:res
keep intact
c. 分解 落实 到
decompose -implement:res -reach:res
break down into
d. 总计 起来
total -up:res
sum up

Verb-resultative compounds are also precisely the class of verbs which can undergo the infixation of 得 de or 不 bu to form verb potential compounds (which are identified as VPT in the Penn Chinese Treebank), a fact reflected neither by the annotation itself nor by the annotation guidelines (Xue et al., 2000).

Many analyses adopt the position that the two branches of the compound are both verbs (Li and Thompson, 1989; Yong, 1997; Packard, 2000). However, Starosta et al. (1998) argues that even when a morpheme used as a resultative compound coincides with a verbal morpheme (e.g. 开 open, -away:res), the two cannot be considered the same morpheme, since in many cases the semantics of the two have diverged in a way which makes the compound non-compositional when taken as a V-V compound.

(74) a. 难 死
difficult -die:res
extremely difficult
b. 叫 哑
shout -mute:res
shout (oneself) hoarse

Although Packard (2000) still considers compounds such as (74-b) as comprised of two verbs, he concludes that while directional resultatives (e.g. 过 -past:res, 进 -in:res) and “stative” resultatives (e.g. 稳 -steady:res, 紧 -tight:res) are identical with their uses as free-
standing verbs, the “attainment” resultatives (e.g. 住 -firm:RES, 见 -perceive:RES) are “bound roots” as opposed to free-standing verbs.

We argue against the analysis of Packard (2000) on the grounds that the stative resultatives are by no means ‘an open and infinitely large class of verbs’. For instance, 稳 -steady:RES, a “stative” resultative according to Packard (2000), is no longer a viable free-standing verb, the disyllabic predicative adjective 稳定 steady having supplanted it in modern Chinese.

We therefore analyse verb-resultatives not as compounds of two verbs, but of one verb and a resultative/directional complement. In our CCG analysis, this results in a head-initial structure where the verb resultative complement is a verbal post-modifier.

(75) The baby cried itself hoarse.

We therefore analyse verb-resultatives not as compounds of two verbs, but of one verb and a resultative/directional complement. In our CCG analysis, this results in a head-initial structure where the verb resultative complement is a verbal post-modifier.

2.9.4 Verb coordination (VCD)

The direct juxtaposition of bare verbs with no intervening coordination word falls under the rubric of the serial verb construction (svc) (Baker, 1989). Because some investigators use the same term to refer additionally to the juxtaposition of VPs, we refer to svc with bare verbs as bare verb coordination.

(76) 西北 首家 乡镇 企业 大厦 建成 开业

northwest first mw township enterprise tower build open.for.business

The first township-level enterprise building in the northwest completes construction and is open for business

In the PCtb annotation, bare verb coordination is represented as a VCD (verb coordination) node, under which lie the bare verbs:
A straightforward CCG representation which preserves this analysis involves the following extra-combinatory rule, which takes any two like verbal categories, and produces the same verbal category.

**Rule 7 (Bare verb coordination rule schema).**

\[
(S[\text{dcl}] \backslash NP) \rightarrow (S[\text{dcl}] \backslash NP)
\]

Because many classes of lexical items in Chinese have verbal senses, any rule which operates on pairs of elements with verbal senses, which include many prepositions (到 to arrive) and nouns (建设 construct ~ construction), Rule 7 proves costly in practice.

For example, each of the word pairs in (78) carry both V and N senses, and hence have the readings V-V, V-N and N-N.

(78)  

a. 批准 同意
    
    permit agree
    
    permit \{ and agree to
    
    consent
    
    permission and consent
    
    b. 推介 宣传
    
    promote advocate
    
    promote \{ and advocate
    
    advocacy
    
    promotion and advocacy

As an alternative to introducing the extra-combinatory Rule 7, Chinese CCGbank considers bare verb coordination to be a morphological operation, the internal structure of which is opaque to syntax. A disadvantage of this approach is that the resulting complexes of multiple serial verbs introduce sparsity.

The distribution of VCD lexical items in the PCTB has a very long tail, with the top ten lexical items accounting for only 64 (5.9%) of the collection of VCD occurrences. However,
we observe that many verb compounds tagged as VCD in the Penn Chinese Treebank have a formulaic character:

(79) a. 利国 利民 利己
benefit-nation benefit-people benefit-self
benefit the nation, the people, and oneself
b. 精干 高效
capable high-efficiency
work at high efficiency

In addition, where the VCD is composed of two verbs, some verbs occur frequently in compound-initial or -final position, the most frequent of which are given in Table 2.10. This suggests that bare verb coordination is not entirely productive, unlike coordination. Additionally, because the VCD prefixes and suffixes in the table occur with high frequency, a tagger should be able to make the determination on the basis of a lexical item's length, prefix and suffix, that a lexical item is a VCD instance and not a free verb. Accordingly we choose the morphologically pre-composed analysis of VCD in which the bare verbs are merged into a single lexical item.

2.9.5 Verb subordination (VSB)

Verb subordination (VSB in the PCTB annotation) differs from bare verb coordination (VCD) in that while the elements in VCD are siblings under an implicit coordination structure, the two verbs in verb subordination exhibit a clear modifier-head relationship, for instance, 介绍 introduce modifying 说 say in (80-a):

(80) a. 介绍 说
introduce say
present, say by way of introduction
b. 出国 度假
leave-country vacation
go on vacation overseas

We suggest that in the same way that VCD should yield the same dependencies as coordination with an overt coordinator, VSB should be analysed in a fashion similar to VP adjunct modifiers:
Chapter 2. The elements of Chinese ccg

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Lexical item</th>
<th>Frequency</th>
<th>Lexical item</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>建成 build</td>
<td>19</td>
<td>投产 put into production</td>
</tr>
<tr>
<td>16</td>
<td>贯彻 carry out</td>
<td>14</td>
<td>建设 build</td>
</tr>
<tr>
<td>13</td>
<td>投资 invest</td>
<td>12</td>
<td>开发 develop</td>
</tr>
<tr>
<td>11</td>
<td>开发 develop</td>
<td>9</td>
<td>落实 carry out</td>
</tr>
<tr>
<td>8</td>
<td>综合 compose</td>
<td>9</td>
<td>实施 bring into effect</td>
</tr>
<tr>
<td>8</td>
<td>组织 organise</td>
<td>8</td>
<td>通车 open to traffic</td>
</tr>
<tr>
<td>8</td>
<td>积极 be active</td>
<td>8</td>
<td>开放 make available</td>
</tr>
<tr>
<td>8</td>
<td>团结 join forces</td>
<td>8</td>
<td>互利 be mutually beneficial</td>
</tr>
<tr>
<td>7</td>
<td>繁荣 prosper</td>
<td>7</td>
<td>编译 translate and edit</td>
</tr>
<tr>
<td>7</td>
<td>平等 be equal</td>
<td>7</td>
<td>稳定 be stable</td>
</tr>
<tr>
<td>7</td>
<td>参观 inspect</td>
<td>7</td>
<td>发展 develop</td>
</tr>
<tr>
<td>6</td>
<td>颁布 distribute</td>
<td>7</td>
<td>兴建 build</td>
</tr>
<tr>
<td>6</td>
<td>研究 research</td>
<td>6</td>
<td>成立 establish</td>
</tr>
<tr>
<td>6</td>
<td>独立 become independent</td>
<td></td>
<td>旅游 travel</td>
</tr>
<tr>
<td>6</td>
<td>扭亏 reverse a loss</td>
<td></td>
<td>执行 carry out</td>
</tr>
</tbody>
</table>

(a) Frequent compound-initial verbs

(b) Frequent compound-final verbs

Table 2.10: Common compound-initial and -final elements and their frequency in Penn Chinese Treebank
2.10. Summary

Combinatory Categorial Grammar analyses of languages such as Arabic, German, Italian, Japanese and Turkish have tested its universality claims, contributing to the literature a better understanding of its strengths and weaknesses, and opening up the possibility of extracting wide-coverage \textit{ccg} grammars for these languages. In this chapter, we have provided the first analysis in the literature of Chinese syntax through \textit{ccg}, covering its basic typology, the structure of NP, coordination, and an account of its verb compounding strategies.

With the analysis of this chapter as a basis, we are now ready to describe how Chinese \textit{ccgbank} accounts for the non-local dependencies arising from constructions such as ex-
traction, topicalisation, pro-drop and wh-movement. Succinct analyses of NLD-projecting constructions are a particular strength of CCG. We show in the next chapter that the machinery of CCG is capable of accounting for the NLD types which are particularly frequent in Chinese syntax.
Chapter 3

The analysis of special syntax

Having developed analyses for the core of Chinese CCG syntax, we focus on a number of constructions which yield non-local dependencies (NLDs). Guo et al. (2007b) observed that while Chinese syntax generates more NLDs than English, few parsers in Chinese are equipped to recover the dependencies arising from NLDs. One of the goals of Chinese CCGbank is to harness the descriptive power of CCG to offer a transparent, concise and consistent account of both local and non-local dependencies. In this chapter, we present the analyses we have developed for a range of Chinese NLD phenomena.

3.1 Control verbs and modal verbs

A control verb subcategorises for an embedded clause whose subject is coindexed with an argument of the control verb. For subject control verbs (e.g. 鼓励 encourage in (1-b)), the subject of the embedded clause is coreferent to the control verb’s subject. For object control
Chapter 3. The analysis of special syntax

verbs (批准 permit in (1-b)), the subject of the embedded clause is instead coreferent to the control verb’s NP complement.

(1)  

a.  

China encourage private entrepreneur invest national basic construction

China encourages private companies to invest in national infrastructure

b.  

nation already permit can establish commercial investment finance  

institutions for commercial investment

Some subject control verbs, like 准备 prepare, subcategorise only for an embedded clause; others, like 鼓励 encourage, subcategorise for an NP in addition to the embedded clause.

According to the Penn Chinese Treebank annotation guidelines, the subject of the embedded clause is an empty category PRO, which is coindexed with the argument to which it is coreferent (Xue et al., 2000). For instance, in the Penn Chinese Treebank annotation, the dependency between 投资 invest and 企业家 entrepreneur is mediated through a PRO-trace appearing in the subject slot of the embedded clause 投资国家基础建设 invest in national infrastructure. However, in the annotation itself, these indices are missing, preventing us from recovering the subject/object control distinction directly from the corpus.

In the ccg analysis, subject and object control, being bounded constructions, project long distance dependencies mediated only by the category of the head word. These dependencies are generated by coindexing the head variables on the parts of the control verb category.

Object control verbs yield the category:

\[ \text{鼓励 encourage} \vdash ((S[dcl] \backslash NP_y) (S[dcl] \backslash NP_z)) / NP_z \]

while subject control verbs yield one of these two categories depending on their subcategorisation:

\[ \text{准备 prepare} \vdash (S[dcl] \backslash NP_y) (S[dcl] \backslash NP_z) \]

\[ \text{批准 permit} \vdash ((S[dcl] \backslash NP_y) (S[dcl] \backslash NP_z)) / NP_y \]

Although modal verbs and two-place subject control verbs project the same dependencies, this fact is obscured in the Penn Chinese Treebank annotation, which treats modal verbs as subcategorising for a VP, due to the underlying theory. In our analysis, modal
verbs receive the same category as two-place subject control verbs:

\[
\text{应当} \quad \text{should} \quad \vdash (S[dcl] \backslash NP_y) / (S[dcl] \backslash NP_y)
\]

### 3.2 Relative clauses and extraction

The CCG analysis of the long-distance dependency generated by extraction from a relative clause demonstrates the re-bracketing facility of the B-T combinatory calculus to assign principled types to non-traditional constituents. We introduce some terminology to simplify the discussion of the relative clause (rc) construction.

![Diagram](adnominal_clause.png)

(2) a. 张三 买的 铅笔

b. pencil, that Zhangsan bought

The Chinese rc construction is head-final, which is consistent with the modification behaviour of other noun-headed compounds, while the English rc construction is head-initial.

The gapping behaviour in gapped extraction is the same in both languages, with the head noun co-referent to a gap in an argument position of the adnominal clause. For instance, in both examples in (2), the head noun (pencil in English; 铅笔 pencil in Chinese) is co-referent to a gap \(t_i\) in an argument position of the adnominal clause (that Zhangsan bought; 张三买 Zhangsan bought). These similarities suggest that the category of the relativiser (的 de in Chinese) or relative pronoun (e.g. that) should be similar modulo directionality.

The English CCG bank relative pronoun category is of the form \((NP \backslash NP) / (S[dcl] \backslash NP)\). Hockenmaier and Steedman (2007) explain that English CCG bank adjoins relative clauses at NP rather than \(N^\prime\) (yielding relativiser categories of the form \((NP \backslash NP) / (S[dcl] \backslash NP)\) rather than \((N \backslash N) / (S[dcl] \backslash NP)\)) for consistency with the (semantically incorrect) annotation in the Penn Treebank. However, as we show, the Chinese relativiser category has the same form, but for a different reason.

In particular, the relative clause construction does not close off a level of \(N^\prime\) in Chinese: of the two modifiers of 铅笔 pencil — the relative clause 张三买的 that Zhangsan bought, and 那支 that mw — the relative clause need not be the outermost.

---

1 "Of these the one which we regret most is that postnominal modifiers such as (restrictive) relative clauses have to be analyzed as NP modifiers, rather than N or N modifiers." (Hockenmaier and Steedman, 2005)
### Table 3.1: Configurations and prevalence of the RC construction in Chinese

<table>
<thead>
<tr>
<th>Relativiser word</th>
<th>Present</th>
<th>Absent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gap</td>
<td>(4) 政府 利用 贷款</td>
<td>(5) 政府 利用 的 贷款</td>
</tr>
<tr>
<td></td>
<td>government use loans</td>
<td>government use de loans</td>
</tr>
<tr>
<td></td>
<td>loans that government uses</td>
<td>loans that government uses</td>
</tr>
<tr>
<td></td>
<td>11128 (39.76%) of sentences</td>
<td>848 (3.03%) of sentences</td>
</tr>
<tr>
<td>Absent</td>
<td>(6) 政府 利用 贷款 情况</td>
<td>(7) 政府 利用 贷款 的 情况</td>
</tr>
<tr>
<td></td>
<td>government use loans situation</td>
<td>government use loans de situation</td>
</tr>
<tr>
<td></td>
<td>the situation of the government using loans</td>
<td>the situation of the government using loans</td>
</tr>
<tr>
<td></td>
<td>4013 (14.34%) of sentences</td>
<td>562 (2.01%) of sentences</td>
</tr>
</tbody>
</table>

Accordingly, the category for the relativiser particle 的 de in the gapped, overt-relativiser case is of the form \((NP/NP)\backslash(S[de]\backslash|NP)\). However, in Chinese, three other variants of the relative clause (RC) construction are also available. Table 3.1 shows the four possible configurations, which arise from the combination of two variables:

**Gapped versus gapless:** whether a gap, coreferent to the head noun, appears in the adnominal clause, and

**Overt versus null relativiser:** whether the relativiser 的 de is present.

Table 3.1 also estimates the prevalence of each configuration in PCTB. To identify candidate instances, we use rules described later in Section 5.4 to retrieve instances of NLD constructions from PCTB for a manual annotation task.

### 3.2.1 Gapless versus gapped 的 de constructions

Chinese syntax permits a relative clause (RC)-like construction in which the head of the RC is not coreferent to any (realised) element of the adnominal clause:
In the gapless case, Li and Thompson (1989) suggests that the head noun and the adnominal clause are in an *about-ness* relation, such that the head noun refers to the instrument, location, time, reason or means by which the action in the adnominal clause takes place. This is consistent with the observation that in the English translations of the RCs in (8), the modification is mediated by prepositions (in and for in the examples).

Because the head noun is not co-referent to any argument position in the adnominal clause, the category of the relativiser does not mediate any long-distance dependency. Accordingly the category of the gapless relativiser in Chinese is \((NP/NP) S[dl]\) — the same form as the gapped relativiser, without any gap in the adnominal clause.

### 3.2.2 Overt versus null relativiser

In English, the RC construction may occur with the relativiser omitted. This configuration is known as a *reduced relative clause* (RRC) in English. In English, RRCs may only occur with object extraction:

\[
(9) \quad \begin{align*}
\text{a. } & \text{The creature, } John \text{ saw } t_i \text{ was black and inky.} \\
\text{b. } & \text{\textasteriskcentered}{The creature, } t_i \text{ fought John was black and inky.}
\end{align*}
\]

While the RRC construction can be handled in pure CCG through categorial ambiguity on nouns or verbs, Hockenmaier and Steedman (2002a) argues convincingly that in the wide-coverage setting, it is unlikely that every lexical item in the training data will appear with its canonical category and also with all the categories required to effect marked syntax such as the RRC construction. Hockenmaier and Steedman (2002a) proposes that type-changing unary rules which rewrite one category as another, are an appropriate compromise in wide-coverage parsing.

Their English CCGbank analysis of the RRC construction in Hockenmaier (2003) involves a type-change rule which performs the same function as the overt relativiser:  

\[\text{Recall that the } \ast \text{-schematisation ranges over a family of modifier categories. } NP_y/\ast \text{ refers to a set }\{NP_y/NP_y, (NP_z/NP_z)_y)/(NP_z/NP_z)_y, \ldots\}\]
Rule 8 (Null relativiser type-change rule).

\[ S[\text{dcl}]|NP_y \rightarrow NP_y \]

Chinese RCs also occur in a configuration without an overt relativiser:

(10) 全省利用外国政府贷款
whole province utilise foreign government loan

foreign government loans used province-wide

The PCTB analysis for this construction is consistent with the overt relativiser case. To maintain the PCTB analysis, we borrow the English CCGbank analysis through type-changing unary rules:

3.2.3 Generalised 的 de modification

The Chinese subordinator 的 de, a homograph of the relativiser 的 de described in the previous section, mediates the modification of a phrasal type by another phrasal type.3

(12) a. 张三 的 手机
Zhangsan’s mobile phone
b. 白色 的 黑板
white-colour blackboard
c. 双人 的 自行车
double-person bicycle

---

3 Li and Thompson (1989) considers the two frames of 的 de to be one and the same, with the relative clause construction being a special case of subordinating a clause to an NP, while the PCTB tagset distinguishes them by giving the generalised subordinator the tag DEG and the relativiser the tag DEC (Xia, 2000a).
3.2. Relative clauses and extraction

d. \([在桌上]_{LCP} 这的杂志_{NP}\)
   at table on LC de magazine
   the magazine on the table

e. \([在北京]_{PP} 这的日子_{NP}\)
   at Beijing de days
   days in Beijing

In the LFG conversion of Guo et al. (2007a), a distinction is made between 的 de indicating possession (12-a) and 的 de indicating modification (12-b), based on the lexical identity of the head of the XP. Guo et al. (2007a) claim that the two can be distinguished on the basis of whether the head of XP in 的 de-modification is a proper noun (PCTB tag NR), to which we propose a counter-example:

\[
(13) \quad 叶问 的 电影
Yip.Man de film
\begin{cases}
   a movie about Yip Man \\
   Yip Man’s movie
\end{cases}
\]

While, in both interpretations, 叶问 Yip Man is a personal name and thus a proper noun (NR), the dependency between it and 电影 film is ambiguously one of modification or possession.\(^4\) Thus, our analysis does not distinguish these two senses of 的 de, because it cannot be recovered from the annotation.

The general analysis for XP de YP modification is to give the subordinator 的 de a category of the form \(YP_y \backslash XP_x) / YP_x\). While this appears similar to the English adjunct preposition category, for instance of \(NP_y \backslash NP_x) / NP_x\), the co-indexation is different, and the Chinese subordinator is much more permissive with respect to the phrasal types it accepts, as demonstrated in Table 3.2.

3.2.4 Pseudo-cleft construction

Chinese has a pseudo-cleft construction which partitions a sentence into presupposition (the background) and focus (new information relating to the background).

\[
(14) \quad 叶宏灯 的 东聚 电业 经常 被 举例
Ye Hongdeng de Dongju Electric often bei raise.example
Ye Hongdeng’s (company) Dongju Electric is often raised as an example.
\]

\(^4\) We note the similarity between the about-ness relation induced by the modification interpretation and that induced by ungapped relative clauses (Section 3.2.1) or S NP apposition (Section 2.4.5).
Chapter 3. The analysis of special syntax

Table 3.2: Types of generalised 的 de modification in PCTB down to frequency 100

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Frame</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>5230</td>
<td>NP de NP</td>
<td>孩子的教育 child’s education</td>
</tr>
<tr>
<td>1538</td>
<td>PP de NP</td>
<td>同以色列的妥协 compromise with Israel</td>
</tr>
<tr>
<td>1173</td>
<td>ADJP de NP</td>
<td>认真的学理研究 serious theoretical research</td>
</tr>
<tr>
<td>1108</td>
<td>QP de NP</td>
<td>76 岁的金大忠 the 76-year-old Jin Dazhong</td>
</tr>
<tr>
<td>1021</td>
<td>LCP de NP</td>
<td>台湾以外的世界 the world aside from Taiwan</td>
</tr>
<tr>
<td>351</td>
<td>NP de QP</td>
<td>过去的一年 the past one year</td>
</tr>
<tr>
<td>226</td>
<td>DP de NP</td>
<td>这一代的孩子 this generation of kids</td>
</tr>
</tbody>
</table>

As reflected by the translation, the Chinese and English pseudo-cleft constructions have similar semantics.

We consider two analyses of the pseudo-cleft construction. The first treats X + 的 de as a nominalisation headed by 的 de, an analysis supported by the distribution of X + 的 de outside of the pseudo-cleft construction:

\[ (15) \quad \text{why \quad do-out:res de compare.to other:people seem fake q} \]

Why is it that what I do seems fake compared to everyone else?

The copula 是 shi then simply receives its canonical category \( (S[dcl]\backslash NP)/NP \):

\[ (16) \quad \text{What is often raised as an example is Ye Hongdeng’s company Dongju Electric.} \]
The category $NP\backslash (S[dcl]|NP)$ in (16) is akin to the usual relativiser category for 的 de, $(NP/NP)\backslash (S[dcl]|NP)$, except for the absence of the RC head. However, we observe that a non-local dependency exists between the clausal gap and the other element in the equative construction, which the above analysis cannot retrieve:

(17) 我们 需要 的 是 和平
1PL need DE is peace
What we need is peace.

We propose a second analysis which treats the combination $NP + 的 de$ in a manner consistent with extraction, so that, as in extraction, the non-local dependency can be retrieved.

(18) 我们 需要 的 和平
1PL need DE peace
the peace that we need

As such, 的 de receives the same category that it does in extraction: $(NP_y/NP_y)/(S[dcl]_z|NP_y)_z$. Then, the copula 是 is receives a category with the coindexation:

$$
\text{是 is } \vdash (S[dcl]\backslash (NP_y/NP_y))/NP_y
$$

This allows the non-local dependency between the clausal gap and the second element in the equative construction to be retrieved:

(19) What we need is peace.

$$
\begin{array}{cccc}
\text{我们} & \text{需要} & \text{的} & \text{是} \\
1PL & need & DE & is \\
NP & (S[dcl]\backslash (NP_y)/NP) & (NP/NP)\backslash (S[dcl]/NP) & (S[dcl]\backslash (NP/NP))/NP \\
S/(S\backslash NP) & \Rightarrow & (S[dcl]/NP) & \Rightarrow \\
NP & S[dcl]\backslash (NP/NP) & NP \\
S[dcl] & \text{NP} & \text{NP} & \text{NP} \\
\end{array}
$$

We propose that where a gap exists and is coreferent to the second element in an equative construction, we adopt the second analysis with the relativiser category for 的 de, and the special coindexation on the copula category. However, where the gap is not coreferent to any element of the sentence, we stipulate that 的 de has the category $NP_y\backslash (S[dcl]_w|NP_z)_w$, so that the result of the combination $X + 的 de$ is of type $NP$, and the gap is unfilled:
Chapter 3. The analysis of special syntax

(20) The ones you make yourself are tastier than the ones you buy.

\[
\begin{align*}
\text{self-made} & \quad \text{compared to} & \quad \text{buy} \\
\text{NP} & \quad \text{VP} & \quad \text{NP}
\end{align*}
\]

\[
\begin{align*}
S[dcl]/NP & \quad \text{NP/}
\end{align*}
\]

\[
\begin{align*}
S[dcl]/NP & \quad \text{NP/}
\end{align*}
\]

\[
\begin{align*}
S[dcl]/NP & \quad \text{NP/}
\end{align*}
\]

\[
\begin{align*}
S[dcl]/NP & \quad \text{NP/}
\end{align*}
\]

\[
\begin{align*}
S[dcl]/NP & \quad \text{NP/}
\end{align*}
\]

3.2.5 On extraction islands

In English, extraction out of a PP complement is grammatical:

(21) the man that I spoke to

\[
\begin{align*}
\text{NP} & \quad \text{NP} & \quad \text{NP} \\
\text{NP} & \quad \text{NP} & \quad \text{NP}
\end{align*}
\]

\[
\begin{align*}
S[dcl]/NP & \quad \text{NP/}
\end{align*}
\]

The Chinese PP is an island, out of which extraction is blocked, rendering the counterpart of (21) ungrammatical:

(22) (intended) *places where pandas live (at)

\[
\begin{align*}
\text{panda} & \quad \text{live at} & \quad \text{place} \\
\text{NP} & \quad \text{NP} & \quad \text{NP}
\end{align*}
\]

\[
\begin{align*}
S[dcl]/NP & \quad \text{NP/}
\end{align*}
\]

\[
\begin{align*}
S[dcl]/NP & \quad \text{NP/}
\end{align*}
\]

\[
\begin{align*}
S[dcl]/NP & \quad \text{NP/}
\end{align*}
\]

\[
\begin{align*}
S[dcl]/NP & \quad \text{NP/}
\end{align*}
\]

\[
\begin{align*}
S[dcl]/NP & \quad \text{NP/}
\end{align*}
\]

\[
\begin{align*}
S[dcl]/NP & \quad \text{NP/}
\end{align*}
\]

\[
\begin{align*}
S[dcl]/NP & \quad \text{NP/}
\end{align*}
\]
To block extraction out of PP\s in Chinese, we stipulate that the category of prepositions such as 在 at has the application-only mode on its slash:

\[
在 \text{ at } \vdash PP/\ast NP
\]

Compared to English, the set of viable candidates for extraction in Chinese is much more limited. Chiu (1995) discusses other positions which constitute islands for extraction, including the inner complement of a ditransitive verb (23-\(a\)) and the complement of 把 ba (23-\(b\)).

\[
(23)\quad \begin{align*}
\text{a. } & *\text{李四 }\text{给 }t\text{ 一 }\text{书 }\text{的 那 个 人} \\
& \text{(intended) the person Lisi gave a book}
\end{align*}
\]

\[
\text{b. } *\text{李四 }\text{把 }t\text{ 打 了 的 那 个 人}
\]

\[
\text{(intended) the person Lisi beat up}
\]

(examples from Chiu (1995))

We account for these two phenomena by placing the application-only mode (\(\ast\)) on the argument-taking slash of 把 ba, and also on the outermost slash of the ditransitive verb category:

\[
\begin{align*}
\text{把 ba } & \vdash (VP/VP_{\ast})/\ast NP \\
\text{DTV } & \vdash (VP/NP)/\ast NP
\end{align*}
\]

The attempted extraction of the recipient argument of 送 give is blocked by the mode on the outermost slash. Analogously, extraction of the 把 ba complement is blocked by the same mechanism.

\[
(24)\quad \text{(intended) } *\text{that person I gave the book to}
\]

\[
\begin{array}{rcccc}
1SG & \text{give} & \text{book} & \text{DE} & \text{that person} \\
NP & (VP/NP_{\text{theme}})/\ast NP_{\text{recip}} & NP_{\text{theme}} & (NP/NP)(S/NP) & NP
\end{array}
\]

\[
\vdash (VP/(VP/NP_{\text{theme}}))^{T}
\]
3.3 Topicalisation and topic-prominence

Many investigators have suggested that Chinese sentences like (25-a) decompose into topic-comment rather than subject-predicate as in languages such as English (Chao, 1968; Chafe, 1976; Li and Thompson, 1989):

(25)  a. 那场火，幸好消防队来得快
     that fire, luckily fire.brigade come DE fast
     That fire, fortunately the fire department came quickly.

     b. 书我买了
        book SFP buy
        As for the book, I’ve bought it.
        (both from Li and Thompson (1989))

The terminology of topic-prominence versus subject-prominence was established by Li and Thompson (1976), who proposed that in topic-prominent languages, sentences like (25-a) are unmarked, or at least less marked, than in subject-prominent languages.

The topic is a sentence-initial constituent with definite reference which acts as background information for the comment (the remainder of the sentence). A sentence may exhibit zero topic, in which case the remainder of the sentence will reflect subject-predicate order. Critically, the topic need not be bound (anaphorically or syntactically) to any element of the comment, unlike subjects, which must necessarily participate in some syntactic relation with the predicate or some element inside the predicate (Li and Thompson, 1976).

For instance, Li and Thompson (1976) claim that in a topic-prominent language such as Chinese, sentences such as (26) exhibit the unmarked sentence order topic-comment, where 这棵 this tree is the topic, and 叶子很大 its leaves are very big is the comment:

(26)  [这棵 this tree]topic, [叶子很大 very big]Comment
      this SFP tree, leaf very big
      As for this tree, its leaves are very big.

For our purposes, we focus on the syntactic distribution of topics, following the definition of ‘topic’ by Her (1991) as a strictly syntactic notion: the displacement of constituents to, or appearance of constituents at, clause-initial (S or VP-initial) position. Huang et al. (2008) notes that displacement-to-S (referred to as “osv structure”) and displacement-to-VP (“sov structure”) place different selectional restrictions on the topicalised constituent.

First, displacement-to-S requires the topicalised constituent to have definite reference:
Displacement-to-VP topicalisation, on the other hand, is free from this constraint:

(28) a. 我 那 部 电影 还 没 看
   1SG that MW film still have-not see
   I still haven’t seen that movie.

   b. 我 一 部 电影 还 没 看
   1SG one MW film even 1SG still have-not see
   I haven't even seen a (single) movie.

The 连 lian construction induces a focusing interpretation with semantics and syntax similar to displacement-to-VP, suggesting unlike displacement-to-S, sov topicalisation may induce a focusing interpretation as well.

(29) 他 连 我 都 不 认识
   3SG lian 1SG also not recognise
   He doesn’t even recognise me.

The second difference is that displacement-to-S topicalisation admits a gapped construction where the gap is coreferent to the topicalised constituent (30-a) as well as a non-gapped construction with a resumptive pronoun coreferent to the topicalised constituent (30-b). In contrast, displacement-to-VP blocks the resumptive pronoun (30-d).

(30) a. 那 部 电影 , 我 还 没 看 t_i
    that MW film 1SG still have-not see

   b. 那 部 电影 , 我 还 没 看 他, t_i
    that MW film 1SG still have-not see 3SG

   c. 我 那 部 电影 , 还 没 看 t_i
    1SG that MW film still have-not see

   d. *我 那 部 电影 , 还 没 看 他, t_i
    1SG that MW film still have-not see
    I still haven’t seen that movie.

The prevalence in PCTB of the three possible configurations of topicalisation is summarised in Table 3.3.
Chapter 3. The analysis of special syntax

Prevalence (% of sentences)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3033</td>
<td>Gapped displacement-to-S</td>
<td>(10.8%)</td>
</tr>
<tr>
<td>714</td>
<td>Non-gapped displacement-to-S</td>
<td>(2.5%)</td>
</tr>
<tr>
<td>12</td>
<td>Gapped displacement-to-VP</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3: Prevalence in PCTB of topicalisation constructions

An CCG analysis of displacement-to-S topicalisation in English, introduced by Steedman (1987) and used in English CCGbank (Hockenmaier and Steedman, 2005), uses the following extra-combinatory unary rule:

**Rule 9** (Unary displacement-to-S gapped topicalisation rule).

\[ X \rightarrow S/(S/X) \text{ for parametrically licensed } X \quad (T_{\text{gap}}) \]

Hockenmaier and Steedman (2005) note that left dislocation (non-gapped displacement-to-S topicalisation) also occurs in English, but Hockenmaier and Steedman choose categorial over derivational ambiguity by requiring the dislocated constituent to carry the category S/S at the top level:

\[
\begin{align*}
(S/S)/N & \rightarrow N, & (S[dcl]\backslash NP)/NP & \rightarrow NP \\
S/S & \rightarrow NP & S[dcl]\backslash NP & \rightarrow S[dcl]
\end{align*}
\]

In Chinese, because topicalisation is much less marked than in English, and the internal structure of the topicalised constituent can grow complex, we choose derivational ambiguity and analyse non-gapped displacement-to-S topicalisation through the following extra-combinatory unary rule:

**Rule 10** (Unary displacement-to-S non-gapped topicalisation rule).

\[ X \rightarrow S/S \text{ for parametrically licensed } X \quad (T_{\text{ngap}}) \]

**Rule 11** (Unary displacement-to-VP topicalisation rule).

\[ X \rightarrow VP/(VP/X) \text{ for parametrically licensed } X \quad (T_{\text{gap}}) \]
3.4. 把 ba and 被 bei constructions

The complex NP topicalisation island effect observed by Huang et al. (2008) which blocks (32) falls out of the condition that forward crossed composition (\(\rightarrow B_{c}\)) is not licensed on the rightward slashes of verbal categories, as in (33):

(32) 李四, 我认识很多喜欢的人  
Lisi, I know many people who \(t_i\) likes \(t_j\).

(33)

\[
\begin{array}{cccc}
NP & NP & VP/NP & (S[dcl]\backslash NP)/\_NP \hspace{1cm} (NP/NP)/(S[dcl]/NP) \\
\_T & NP & S/V \_P & NP \backslash (S[dcl]/NP) \\
S/(S/NP) & S[dcl]/NP & B & B \hspace{1cm} B \backslash (S[dcl]/NP) \\
\end{array}
\]

3.4 把 ba and 被 bei constructions

把/被 ba/bei are particles which introduce two related constructions which both vary the canonical svo word order of Chinese sentences: 被 bei raises the direct object to a subject position (34-b), while 把 ba raises the direct object to a focused pre-verbal position (35-b).

(34) a. 我发那封信送给你  
1SG send that MW letter send-to 2SG

b. 我把那封信送给你  
1SG BA that MW letter send-to 2SG

I sent that letter to you.

(35) a. 他抢走了我的钱包  
3SG snatch-away ASP 1SG DE wallet  
He snatched my wallet.

b. 我的钱包被他抢走了  
1SG DE wallet BEI 3SG snatch-away ASP  
I had my wallet stolen.

The 被 bei construction has been studied at length in the Chinese generative linguistics literature, because while it appears to satisfy the same role as the English be-passive, its distribution and syntax is demonstrably different. Similarly to the Japanese に ni-adversative passive, the 被 bei passive imubes its subject complement with an “adverseness” reading, where it is negatively affected by the NP complement of 被 bei (the agent) performing the
action of 被 bei's embedded clause. For instance, in (36), 城市 the city is adversely affected by 雨 rain falling.

(36) 城市 被 雨 困住
city bei rain imprison-firm
The city was besieged by rain.

On the other hand, the 把 ba construction imbues the syntactic direct object of its embedded clause with a “disposal” reading (Chao, 1968), where the NP complement of 把 ba is affected by the event described in the embedded clause. For instance, in (37), the complement of 把 ba, 程序 program is affected by the action represented by 贴 paste.

(37) 我 把 程序 贴出来 , 大家 帮 我 看看
1SG BA program paste-up:RES , everybody help 1SG look-REDUP
I've posted the computer program up — everybody take a look at it for me.

The prevalence of the 把 ba and 被 bei constructions in PCTB is summarised in Table 3.4. The prevalence of passivisation in English, particularly in formal written registers, is very high — in English CCGbank, 11965 (24.45%) of sentences contain a passivisation.5 In light of this, it may be surprising that the productivity of passivisation in PCTB is so low. However, Chinese has alternative strategies which achieve a similar goal to English passivisation, such as the notional passive of Yip and Rimmington (1997) or the middle construction of Ting (2006), which achieve a passive reading without using either form of 被 bei:

(38) 城市 已经 炸毁 了
city already bomb-destroy:RES ASP
The city has already been destroyed by bombs.

Despite being relatively infrequent, the 把/被 ba/bei constructions are still well-enough attested in the treebank, and certainly well-enough attested in the Chinese syntax literature, that we develop a CCG analysis which draws on existing approaches from the generative syntax literature.

We first examine the prevailing analyses of the 被 bei construction, as its distribution is more complex.

---

5 This includes post-nominal modifiers such as expertise backed by a qualification.
3.4. 把 ba and 被 bei constructions

<table>
<thead>
<tr>
<th>Prevalence Type</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>471 (1.68%)</td>
<td>Long 被 bei</td>
</tr>
<tr>
<td>887 (3.17%)</td>
<td>Short 被 bei</td>
</tr>
<tr>
<td>1155 (4.13%)</td>
<td>把 ba</td>
</tr>
</tbody>
</table>

Table 3.4: Prevalence in pctb of 把/被 ba/bei constructions

3.4.1 The long 被 bei construction

The 被 bei construction comes in two configurations: long (39-a) and short (39-b).

(39) a. 小偷 被 警察 逮捕 了
thief 被 police arrest ASP
The thief was arrested by the police.
b. 小偷 被 逮捕 了
thief 被 arrest ASP
The thief was arrested.

As Huang et al. (2008) note, a naïve analysis would be to consider the two configurations identical modulo the specification of an optional agent NP, just as the by-PP in the English be-passive is not obligatory. However, syntactic, semantic and diachronic evidence argue against this analysis.

In long 被 bei (39-a), the NP complement of 被 bei is coindexed with the subject of its VP complement, forming a long-distance dependency.

The literature diverges on whether 被 bei triggers A-movement of the embedded clause’s direct object to subject position as in English (the movement position), or is simply a lexical verb which subcategorises for two arguments: an agent, and a gapped sentence (the complementation position). A third, more recent analysis by Feng (1997) accounts for many of the issues which affect the movement and complementation approaches by proposing that elements of both analyses take place.

CCG, as a trace-free formalism, does not use movement to account for syntactic constructions where constituents are displaced from their canonical position. Similarly, the LFG account of 被 bei in Guo et al. (2007a) relies on re-entrancy instead of traces to express the non-local dependency between the gap and a verbal argument. Nevertheless, movement analyses in generative syntax have lexically specified parallels in CCG. We will
examine the claims of the movement and complementation hypotheses in terms of their impact on the resulting CCG analysis.

The movement hypothesis

The movement hypothesis proposes the following S-structure for long 被 bei:

\[(40)\]

\[
\begin{array}{c}
\text{NP} \\
\text{张三 Zhangsan}_i
\end{array}
\] \quad \begin{array}{c}
\text{VP} \\
\text{PP} \\
\text{张三 Zhangsan}_i \quad \text{bei Lisi} \quad \text{打了 hit} \\
\text{VP} \\
\text{张三 Zhangsan}_i \quad \text{bei Lisi} \quad \text{打了 hit} \\
\text{S}\llbracket dcl \rrbracket \\
\text{NP} \\
\text{张三 Zhangsan}_i \quad \text{bei Lisi} \quad \text{打了 hit} \\
\text{NP} \\
\end{array}
\]

In this analysis, the agent argument (李四 Lisi) becomes the complement of 被 bei, forming a PP adjunct, and the object NP of the embedded verb 打 hit (张三 Zhangsan) undergoes movement to the subject position, whose agent theta-role has been suppressed under 被 bei passivisation.

A CCG analysis which preserves the structure of (40) involves the following long 被 bei category, which we abbreviate for space in the discussion to come:

被 bei ⊨ ((S[dcl]\NP_z)/(S[dcl]\NP_y)/\NP_y)/\NP_y
≡ ((S[dcl]\NP_z)/TV_{y,z})/\NP_y

The following example demonstrates an analysis using this category.

\[(41)\] Zhangsan was hit by Lisi.

\[
\begin{array}{c}
\text{NP} \\
\text{张三 Zhangsan} \\
\text{bei Lisi} \quad \text{打了 hit} \\
\text{ASP}
\end{array}
\]

\[
\begin{array}{c}
\text{NP} \\
\text{张三 Zhangsan} \\
\text{bei Lisi} \quad \text{打了 hit} \\
\text{ASP}
\end{array}
\]

\[
\begin{array}{c}
\text{NP} \\
\text{张三 Zhangsan} \\
\text{bei Lisi} \quad \text{打了 hit} \\
\text{ASP}
\end{array}
\]

\[
\begin{array}{c}
\text{NP} \\
\text{张三 Zhangsan} \\
\text{bei Lisi} \quad \text{打了 hit} \\
\text{ASP}
\end{array}
\]
3.4. 把 ba and 被 bei constructions

Contrary to the $X'$ movement analysis, we cannot preserve the stipulation that 被 李四 bei Lisi forms an *adjunct* of type PP — in ccg, all long distance dependencies are projected by lexicon entries, and we cannot predict the necessary long distance dependencies when 被 李四 bei Lisi is analysed as an (optional) pre-verbal modifier.

Luckily, the prediction of the $X'$ movement analysis that 被 李四 bei Lisi forms a PP is one of its downfalls — Bei + NP does not follow known patterns of PP distribution such as movement to S-initial position:

\[(42)\]

a. 对 那件事 她 很 认真  
   towards this Sg matter 3Sg very serious
b. 她 对 那件事 很 认真  
   3Sg towards this Sg matter very serious
   She is very serious about this matter.

\[(43)\]

a. 他 被 李四 骂 了 一顿  
   3Sg bei Lisi scold ASP one spell
b. *他 被 李四 骂 了 一顿  
   bei Lisi 3Sg scold ASP one spell
   (intended) He got told off by Lisi.

Therefore, it is neither possible nor desirable to maintain the PP adjunct nature of the movement analysis.

The complementation hypothesis

Another major class of analyses proposes that 被 bei is a lexical verb which denotes to undergo, to experience, and subcategorises for a gapped S containing the agent NP and the action VP (44). This analysis supposes that the interpretation for (39-a) is: The thief experienced the event: “The police arrested the thief.” and that the occurrence of the thief in the embedded clause is deleted under identity with the subject of 被 bei.
Because the absence of the embedded clause's object in (44) is due to deletion and not movement, it is a pro trace (Huang et al., 2008). At the same time, Huang et al. (2008) suggests two problems with treating this as pro-drop— the evidence for object pro-drop in Chinese is inconclusive, and that contrary to instances of pro-drop, the absence of the object is obligatory.

An alternative analysis by Feng (1997) suggests that 被 bei subcategorises for an S, whose embedded direct object undergoes movement to Spec of S. 被 bei's syntactic subject then binds the moved direct object of the embedded clause of 被 bei.
Under the analysis of Huang et al. (2008), 被 bei subcategorises for a full sentence which undergoes object pro-drop. The CCG reflex of this analysis is to assign 被 bei the category \((S[dcl]\backslash NP)/S[dcl]\), where the object of the \(S[dcl]\) complement undergoes pro-drop. Because pro-drop is known to be optional, in contrast to the object gap position in 被 bei which is obligatory, we consider this analysis a poor candidate, as it cannot explain the conditions under which the object pro-drop occurs.

Under the analysis of Feng (1997), 被 bei subcategorises for an object-gapped constituent (the fact that the trace is moved to Spec of IP is a theory-internal detail which does not influence the CCG analysis). The CCG analogue of this analysis therefore treats the object-gapped \(S\) as just that: a constituent of category \(S[dcl]/NP\), rather than a constituent of category \(S[dcl]/NP\) which undergoes pro-drop to category \(S[dcl]\). This analysis predicts that the \(S\) complement of 被 bei is obligatorily object-gapped:
Chapter 3. The analysis of special syntax

The earring got dropped by me.

The earring got dropped by me.

NP (S[dcl]\NP)/(S[dcl]/NP)

(S[dcl]\NP)/NP (S\NP)/(S\NP)

NP

S/(S\NP)

S[dcl]/NP

S[dcl]/NP

S[dcl]

We summarise the CCG analogues of each of the above three long 被 bei analyses in Table 3.5. Firstly, we reject the analysis of Huang et al. (2008), since the supposition that the object gap is pro-drop does not adequately explain its obligatory nature. Furthermore, under the analysis of Huang et al. (2008), there would be no way to recover the dependency between the object gap and the subject of 被 bei, since 被 bei's complement S[dcl] would be considered saturated following pro-drop.

The analysis in Feng (1997) has an advantage which will not be obvious until we consider the CCG analysis of the related 把 ba construction — the complementation analysis of 被 bei yields a category of 被 bei which is structurally identical to the the category of 把 ba, modulo head co-indexation. The implementation of categories in the C&C parser assumes that each category string is mapped to a single head co-indexation. Although we describe a method in Chapter 6 to distinguish head co-indexations which happen to share a structural category, we propose adopting the analysis of Feng (1997) to simplify the act of distinguishing its unique head structure.

3.4.2 The short 被 bei construction

The short 被 bei superficially resembles the long 被 bei construction with the agent NP (the complement of 被 bei) deleted:

a. 张三 被 李四 打
   Zhangsan 被 Lisi 打
   Zhangsan got hit by Lisi.

b. 张三 被 打
   Zhangsan 被 打
   Zhangsan got hit.
Table 3.5: Category of 被 bei in the ccg analogues of the three analyses of long 被 bei

<table>
<thead>
<tr>
<th>Work</th>
<th>Category of long 被 bei</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gapped</td>
</tr>
<tr>
<td>Feng (1997)</td>
<td>(S[dl] \ NP) / (S[dl] / NP)</td>
</tr>
</tbody>
</table>

Li and Thompson (1989) argues that the short 被 bei construction cannot be derived from the long 被 bei construction by deletion, regardless of whether one adopts the complementation hypothesis or the movement hypothesis. If 被 bei is analysed as a preposition (the movement hypothesis), deletion of its complement cannot occur as preposition stranding is not possible in Chinese. If 被 bei is analysed as subcategorising for an agent NP and a verb, movement out of that argument position is not possible either. For example, consider the causative verb 使 make. If 被 bei were analysed as a verb, its subcategorisation frame would coincide with that of 使 make: namely, V NP VP. However, extraction out of the NP argument position of 使 make is impossible:

(48) *我 使 做饭 的 人
     I make cook.a.meal de person
     (intended) the person I made cook a meal

Therefore, short 被 bei cannot be derived from the analysis of 被 bei as a lexical verb, since movement out of the NP argument position of a lexical verb with the subcategorisation frame V NP VP is not possible in Chinese.

Huang et al. (2008) instead suggests that short 被 bei is an instance of A’-movement, namely movement of the trace to PRO at Spec of VP, which is then controlled by the matrix subject:
The CCG analogue of this analysis is a category of 被 bei which subcategorises for a transitive verb, with a head coindexation which coindexes the matrix subject with the patient of the transitive verb.

$$\langle \text{gapped short 被 bei} \rangle \vdash (S[\text{dcl}]\backslash NP_p)/(S[\text{dcl}]\backslash NP)/NP_p$$

(50) Zhangsan got hit.

This coindexation of gapped short 被 bei is isomorphic to the LFG f-structure for the same construction induced by Guo et al. (2007a). Furthermore, our analysis agrees with Huang et al. (2008)’s observation that the Chinese short passive has an analysis isomorphic to English tough-movement; our category for short-bei is the same as Hockenmaier and Steedman (2005)’s category for English tough-adjectives.

In the non-gapped configuration, no element in the VP complement is co-referent with the matrix subject:
3.4. 把 ba and 被 bei constructions

(51) 张三 被 打断 了一 条 腿
Zhangsan bei hit-break:res asp one mw leg
Zhangsan got his leg broken.

such that the coindexation of non-gapped short 被 bei is:

\[ \langle \text{non-gapped short 被 bei} \rangle \vdash (S[dcl]\backslash NP_p)/(S[dcl]\backslash NP) \]

3.4.3 The 把 ba construction

A variety of analyses of the 把 ba construction exist in the literature which echo the analyses of the long 被 bei construction, including an analysis of 把 ba as a preposition (Li and Thompson, 1989), as a lexical verb (Hashimoto, 1971), and as a case marker (Huang, 1982a).

The characteristics of the 把 ba construction are the raising of a patient argument to a pre-verbal position, introduced by a particle 把 ba, as well as imbuing the patient with a “disposal” reading (Chao, 1968; Li and Thompson, 1989). For instance, the patient argument 垃圾 rubbish in the canonical frame becomes the complement of a particle 把 ba in the 把 ba construction:

(52) a. 我扔掉 垃圾
1SG throw-away:res asp rubbish
I threw the rubbish away.

Unlike the 被 bei construction, the 把 ba construction does not occur in a “short” configuration. However, both the 把 ba and long 被 bei constructions occur in gapped and non-gapped frames. In the gapped configuration of 把 ba, as in the 被 bei construction, the complement of the particle 把/被 ba/bei is co-referent with a gap in the clausal complement:

(53) 我 把 垃圾 扔掉 了 ti
1SG ba rubbish throw-away:res asp
I threw the rubbish away.

* We consider the 把 ba-frame (52-b) to be less marked than the canonical frame (52-a). Yang (2008) attributes this difference in acceptability to the cross-linguistic phenomenon of Differential Object Marking (DOM), in which syntactic objects are marked according to criteria such as definiteness, animacy and specificity. In the DOM framework, the object 垃圾 rubbish is non-specific (the sentence does not presuppose the existence of a particular piece of rubbish) and inanimate, which causes (52-b), in which the object is case-marked with 把 ba, to be preferred (Yang, 2008).
In the non-gapped configuration, there is no gap in the clausal complement, but an “about-ness” relation holds between the complement of 把 ba and the action in the clausal complement, reminiscent of the relation between the head noun and the adnominal clause in the non-gapped relative clause construction:

(54) 把目光投向香港
BA attention throw-facing:res Hong Kong
focusing attention on Hong Kong

The same observation which led us to reject the analysis of Bei + NP as a phrasal unit in the 被 bei construction leads us to reject the corresponding analysis for the 把 ba construction, given the unnaturalness of:

(55) *我把垃圾和白纸扔在一起
1SG BA rubbish and BA paper throw at together
(intended) I took the rubbish and the paper and threw them together.

We propose the following category assignments for the two configurations of 把 ba, which echo the structural category of the analysis through complementation of the long 被 bei construction:

\[
\begin{align*}
\langle \text{gapped 把 ba} \rangle & \vdash ((S[dcl]\backslash NP_a)/TV_{a,p})/NP_p, \\
\langle \text{non-gapped 把 ba} \rangle & \vdash ((S[dcl]\backslash NP_a)/(S[dcl]\backslash NP_a))/NP
\end{align*}
\]

### 3.4.4 Summary of the 把/被 ba/bei constructions

We have accounted for the six configurations of the 把 ba and 被 bei constructions in Chinese — the gapped and non-gapped 把 ba construction, and the gapped and non-gapped variants of the long and short 被 bei constructions. The categories and coindexations for each configuration of the 把/被 ba/bei constructions are reproduced in Table 3.6.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Gapped</th>
<th>Non-gapped</th>
</tr>
</thead>
<tbody>
<tr>
<td>long 被 bei</td>
<td>$(S[dcl]\backslash NP_p)/(S[dcl]/NP_p)$</td>
<td>$(S[dcl]\backslash NP_p)$</td>
</tr>
<tr>
<td>short 被 bei</td>
<td>$(S[dcl]\backslash NP_p)/(S[dcl]\backslash NP_p)$</td>
<td>$(S[dcl]\backslash NP_p)/(S[dcl]\backslash NP)$</td>
</tr>
<tr>
<td>把 ba</td>
<td>$(S[dcl]\backslash NP_a)/TV_{a,p})/NP_p$</td>
<td>$(S[dcl]\backslash NP_a)/(S[dcl]\backslash NP_a))/NP$</td>
</tr>
</tbody>
</table>

Table 3.6: Summary of 把/被 ba/bei categories
3.5 pro-drop

Pro-drop, the omission of pronouns in argument positions, is grammatical in some languages (e.g. Turkish, Italian, Arabic, Japanese, and Mandarin Chinese) yet ungrammatical in others (e.g. English, French, Dutch, and German) (Zwart, 1997; Neellemen and Szendrői, 2007). Those languages which license some form of pro-drop may place various constraints on which argument positions license pro-drop — Italian blocks pro-drop of both oblique pronouns and object pronouns (Neeleman and Szendrői, 2007), while Arabic only allows pro-drop of the subject (Boxwell and Brew, 2010).

An early and influential attempt to characterise those languages which allow pro-drop was Taraldsen (1980), whose generalisation suggested that it is languages with rich verbal agreement which tend to license pro-drop, because in such languages, the information in the covert argument can be recovered from the agreement features in the verb.

Taraldsen’s generalisation also predicted that zero-agreement languages like Chinese, Japanese and Korean would block pro-drop, when in fact, the data suggest that these languages pro-drop enthusiastically and in a less constrained manner than in Romance-style pro-dropping languages such as Italian. This kind of pro-drop, which is licensed on any argument of any verb, is sometimes referred to as radical pro-drop (Neeleman and Szendrői, 2007). We estimate the prevalence of pro-drop in the PCTB by tabulating the number of times each pro-drop rule is attested in Chinese CCGbank. This is because the subcategorisation frames of pro-dropped verbs are not straightforward to recover directly from the PCTB annotation.

The frequency of the rules is given in Table 3.7. We also tabulate the most frequently pro-dropped verbs in Table 3.8, showing that while pro-drop is licensed on all verbs, certain verbs exhibit it more often.
Chapter 3. The analysis of special syntax

<table>
<thead>
<tr>
<th>Total (% pro)</th>
<th>Verb</th>
<th>Total (% pro)</th>
<th>Verb</th>
</tr>
</thead>
<tbody>
<tr>
<td>1064 (26.7%)</td>
<td>有 have</td>
<td>167 (75.0%)</td>
<td>请 ask</td>
</tr>
<tr>
<td>662 (8.2%)</td>
<td>是 is</td>
<td>155 (20.8%)</td>
<td>让 allow, cause</td>
</tr>
<tr>
<td>436 (24.3%)</td>
<td>要 need</td>
<td>149 (32.4%)</td>
<td>希望 hope</td>
</tr>
<tr>
<td>234 (26.7%)</td>
<td>可以 can</td>
<td>121 (3.9%)</td>
<td>说 say</td>
</tr>
<tr>
<td>196 (13.4%)</td>
<td>会 will</td>
<td>113 (8.0%)</td>
<td>被 bei</td>
</tr>
<tr>
<td>182 (19.0%)</td>
<td>能 can</td>
<td>104 (21.8%)</td>
<td>可 can</td>
</tr>
<tr>
<td>182 (24.9%)</td>
<td>没有 does not have</td>
<td>99 (59.6%)</td>
<td>预计 predict</td>
</tr>
</tbody>
</table>

Table 3.8: Most frequently pro-dropped verbs in PCTB down to frequency 99. (% pro) shows the relative frequency of pro-drop for each verb to all occurrences.

This apparent disparity was taken up by Huang (1984), who distinguished between “hot” pro-drop-blocking languages such as English (56), “medium” pro-drop languages such as Italian (57), and “cool” pro-drop languages such as Chinese (58):

(56) a. She saw John.
     c. *John knows that Mary has seen.

(57) a. Lei ha visto Gianni.
     3SG.F have:3SG.PRES see:PPT.SG.M Gianni.
     She saw Gianni.
     b. Ha visto Gianni.
     have:3SG.PRES see:PPT.SG.M Gianni.
     (She) saw Gianni.
     c. *Gianni sa che Maria ha visto.
     Gianni know:3SG.PRES that Maria have:3SG.PRES see:PPT.SG.M
     *Gianni knows that Maria has seen (him).

(58) a. 她看见李四了
     3SG see:see:RES Lisi SFP
     She saw Lisi.
     b. 看见李四了
     see:see:RES Lisi SFP
     (She) saw Lisi.
     c. 张三知道李四看见了
     Z. knows L. see SFP
     Zhangsan knows that Lisi saw (him).
d. 张三说 pro 不认识 李四
Z. says pro not know L.
Zhangsan says (he) does not know Lisi.

In Huang (1984)'s analysis, what is licensed by “cool” languages such as Chinese is topic-drop, where a discourse topic is raised to sentence-initial position and subsequently dropped from topic position. True pro-drop, which is the only kind licensed by “medium” languages like Italian, is only grammatical when the empty pronominal can bind either to an agreement node (Agr) or a referent NP (which Huang (1984) calls the Generalised Control Rule or gcr). Thus, two characteristics of Chinese-style pro-drop are explained: 1) it is more permissive with respect to what arguments may be dropped, because the set of potential candidates is constrained by discourse instead of syntax; and 2) because Chinese does not exhibit agreement, the only environment which satisfies the gcr is when the subject of an embedded clause is coreferent with the matrix subject, as in (58-d).

According to Huang (1984), “cool” languages license the dropping of any argument also licensed for topicalisation, including objects, unlike “medium” languages in which pro-drop is conditioned on properties of the verb (for instance, agreement features in Italian).

The implications of Huang’s proposal on our ccg transfer are that: 1) the viability of argument dropping in Chinese is not conditioned on the verb — all verbs are candidates, and 2) any argument of any verb, not just subjects, may be dropped.

The first point suggests that a transfer which only predicts argument dropping on particular verbs will under-generate, and the second point suggests that the chosen mechanism for accounting for argument dropping must be able to delete any argument of any verb.

Finally, Levy and Manning (2003) also notes that pro-drop causes the following ambiguity between VP coordination and IP coordination, because a pro-dropped IP has the same realisation as a VP.

![Diagram](image)

Topicalised constituents can also interact with pro-drop to generate derivational ambiguity:
Chapter 3. The analysis of special syntax

3.5.1 Three approaches to modelling pro-drop

The three approaches to modelling pro-drop with CCG are categorial ambiguity (Çakıcı, 2008), lexical rules (Boxwell and Brew, 2010) and unary rules (Bozsahin et al., 2006).

Categorial ambiguity

The categorial ambiguity approach assigns distinct lexical categories for a verb depending on what arguments it will drop. For example, a Chinese subject-dropped transitive verb will have category $S[\text{dcl}]/NP$ in addition to its canonical category $(S[\text{dcl}\setminus NP])/NP$. This approach will induce considerable data sparsity: a verb will never receive a pro-dropped category if it has never been attested in a pro-drop construction in the corpus. A related issue is that under this analysis, in languages which license pro-drop on any argument of any verb, there will be $2^n$ categories for each verb of $n$ arguments, because this analysis for pro-drop cannot drop arguments incrementally.

Lexical rules

The lexical rules approach generates derived lexical entries for pro-dropped verbs from the canonical lexical entries for verbs, viz.:

$$(S[\text{dcl}\setminus NP]_1 \Rightarrow S[\text{dcl}]_2)$$

While we can use this approach to expand the lexicon, it is not guaranteed that the generated derived lexical entries will each occur in the corpus. For instance, a given verb may never occur pro-dropped in the corpus despite being licensed for pro-drop in a given language. As such, models trained on the corpus will not be able to infer the distribution of generated lexical entries. A technique which can accommodate generated lexical rules is to heuristically transform existing Treebank derivations into instances containing the derived lexical entries. For instance, to learn a distribution of pro-dropped verb categories in Chinese, we could generate a corpus which randomly replaces the categories of verbs in non-pro-drop sentences with their pro-dropped counterparts. However, this approach is
unsatisfactory because the distribution and viability of pro-drop is governed by semantic and pragmatic principles, even if pro-drop is syntactically licensed on all arguments of all verbs in a radical pro-drop language such as Chinese. As a consequence, the generated sentences may be syntactically viable but semantically implausible.

Unary rules

The unary rules approach\(^7\) allows verb categories to be rewritten as their pro-dropped counterparts at parse-time:\(^8\)

Rule 12 (Subject-dropping rule).

\[(S[\text{dcl}] \backslash \text{NP})/\text{NP} \rightarrow S[\text{dcl}]\]

Using unary rules increases parser ambiguity, as the parser must now generate pro-dropped categories whenever it considers a canonical verbal category. Furthermore, an analysis of pro-drop through unary rules is only appropriate in non-radical pro-drop languages (the “medium” languages of Huang (1984)), because it is predicated on the fact that radical pro-drop is licensed on any and all verbs. However, the unary rules approach allows a statistical parser to infer a model from training data of when pro-drop is appropriate, accounting for a flaw of the aforementioned scheme which generates pro-dropped sentences from canonical treebank sentences.

On the basis that Chinese exhibits radical pro-drop, and because of the limited size of the source corpus, we choose the analysis through unary rules of pro-drop for Chinese CCGbank.

---

\(^7\) Boxwell and Brew (2010) point out that in Arabic, only a nominative-case argument may be pro-dropped. However, because the category of a pro-dropped transitive verb coincides with the category of a non-pro-dropped intransitive verb, features must be used to prevent pro-drop from operating on both of the arguments of a transitive verb. This approach is also used by Çakıcı (2008) in Turkish, where \(S[\text{nom}]\backslash \text{NP}\) is the category of an intransitive verb or an object-pro-dropped transitive verb, and \(S\backslash \text{NP}\) is the category a subject-pro-dropped transitive verb.

In a non-verb-initial language such as Chinese, as long as no object-pro-drop rule of the form \(S[\text{dcl}]\backslash \text{NP} \rightarrow S[\text{dcl}]\) exists, no such constraint is necessary to prevent pro-drop from applying twice, because the directionality of the subject- and object-consuming slashes differs.

\(^8\) More generally, we can specify that the subject of any verb may be dropped regardless of its arity using a rule schema:

\[(S[\text{dcl}]\backslash \text{NP})_{\text{s}_1} \rightarrow S[\text{dcl}]_{\text{s}_1}\]
Table 3.9: Distribution of non-VP predicate types pctb down to threshold 100

3.6 Null copula

While attributive sentences lacking an overt copula can be analysed as having a null copula, we can also consider Chinese as a language which permits constituents apart from VPs to form complete predicates by themselves. The distribution of non-VP predicate types in pctb is given in Table 3.9.

In CCG, all predicates must possess a category of the form S[X]/NP, regardless of their syntactic type. This is a prime example of a form-function distinction as targeted by Honnibal (2010); for example, while a QP predicate has the form of a QP, its function in a null copula sentence is that of a predicate. Thus, under Honnibal’s hat-ccg, a QP predicate would receive the category QP^[def]/NP.

Arabic also exhibits zero copula when equating an NP with an NP (Ryding, 2005, p. 319). However, the construction differs in that an echo pronoun can intervene between the two NPs, essentially serving as a copula:
In this construction, the third person pronoun *huwa* could be treated as a copula, giving it the copula category (*S*[dcl]\*NP)/NP despite its sense as a pronoun.\(^9\) However, no such word mediates the zero copula construction in Chinese, and indeed, an explicit copula is ungrammatical, except in a marked context where the predicate gains a focusing interpretation:

(65)  *她 是 十八 岁*

3SG is 18 years:MW

*(intended)* She is 18 years old.

Accordingly, when hat-CCG is not available, all predicates must have CCG category *S*[dcl]\*NP, regardless of their form, which unavoidably engenders lexical ambiguity and modifier proliferation:

(66)  **He is 18 years old.**

他 十八 岁

3SG eighteen MW

NP \(\rightarrow\) \((S[dcl]\*NP)/M\) \(\rightarrow\) \(M\)

\(\rightarrow\) \(S[dcl]\*NP\)

\(\rightarrow\) \(S[dcl]\)

### 3.7 *wh*-questions

The extraction-like fronting phenomena exhibited by English *wh*-questions are distinctively absent in Chinese, a state of affairs known as *wh*-in-situ.\(^{10}\)

(67)  a. **麒 麟 飞 到 北 极 变 什么?**

Qilin fly-reach:res north.pole become what?

What do you call a qilin who flies to the north pole?

b. **麒 麟 飞 到 北 极 变 冰淇淋**

Qilin fly-reach:res north.pole become ice.cream

\(^9\) This solution was suggested by Stephen Boxwell (p.c.) in reference to an Arabic CCG grammar.

\(^{10}\) Particular types of *wh*-questions in English — echo questions, multiple *wh*-questions and declarative questions — cause *wh*-in-situ (Trotta, 2000), but the canonical single-*wh* information question results in characteristic *wh*-movement.
A qilin who flies to the north pole becomes ice cream.

(冰麒麟 ice qilin is homophonous with 冰淇淋 ice cream)

(68) a. What do you get if twenty rabbits step backwards?
b. You get a receding hare line.

In languages which exhibit \textit{wh}-in-situ, the word order of a declarative sentence (67-b) is the same as one where a \textit{wh}-word asks after one of its arguments (67-a). In contrast, in the surface syntax of languages with \textit{wh}-fronting, the \textit{wh}-word appears in sentence-initial position, and a trace appears in the \textit{wh}-phrase which is co-referent with a \textit{wh}-operator.

That \textit{wh}-in-situ languages do not invoke the movement machinery of \textit{wh}-fronting languages was a viewpoint challenged early on by Huang (1982b), who argued that \textit{wh}-words are at Spec of CP in LF (logical form) even in overt \textit{wh}-movement languages, and it is the \textit{wh}-in-situ languages where movement occurs from this position to yield the surface syntax.

While the two concepts of movement and separation of logical and surface form implicit in Huang (1982b)’s proposal both go against the theoretical grain of \textit{ccg}, it nevertheless has an important implication on our \textit{ccg} transfer: even in a \textit{wh}-in-situ language like Chinese, the \textit{wh}-word must be at a position where it can command the rest of the sentence. The \textit{ccg} analogue of this requirement is that the \textit{wh}-word, which triggers the special argument-taking behaviour, be the lexical item which “powers” the construction.

The freer concept of constituency in \textit{ccg} enables precisely such an analysis, as exploited by Hockenmaier and Steedman (2002b) and Clark et al. (2004) in English, where the derivation is “powered” by the category of the \textit{wh}-word $S[wq]/(S[q]/NP)$ (for object extraction) or $S[wq]/(S[dc]NP)$ (for subject extraction).

\begin{itemize}
\item \textbf{a.} Who did he recommend?
\item \textbf{b.} You get a receding hare line.
\end{itemize}
b. Who did Zhangsan recommend?

Who: $\text{谁 (who)}$
Recommend: $\text{推荐 (recommend)}$
Zhangsan: $\text{张三 (Z.)}$

```
NP \rightarrow (S[dcl]\NP)/NP \rightarrow (S'\NP)\(S'\NP) \rightarrow S[wq]\(S[dcl]\NP)
\rightarrow (S[dcl]\NP)/NP
\rightarrow S[dcl]\NP
\rightarrow S[wq]
```

(70) a. Who recommended him?

Who: $\text{谁 (who)}$
Recommend: $\text{推荐 (recommend)}$
Him: $\text{他 (he)}$

```
S[wq]/(S[dcl]\NP) \rightarrow (S[dcl]\NP)/NP \rightarrow NP \rightarrow S[dcl]\NP \rightarrow S[wq]
```

b. Who recommended Zhangsan?

Who: $\text{谁 (who)}$
Recommend: $\text{推荐 (recommend)}$
Zhangsan: $\text{张三 (Z.)}$

```
NP \rightarrow (S[dcl]\NP)/NP \rightarrow (S'\NP)\(S'\NP) \rightarrow S[wq]\(S[dcl]\NP)
\rightarrow (S[dcl]\NP)/NP
\rightarrow S[dcl]\NP
\rightarrow S[wq]
```

The *wh*-word may also appear in two PP positions: inside a PP adjunct (71-a), or inside a PP complement (71-b):

(71) a. 你 刚 跟 谁 说话？
      you just with who speak?
      Who were you just speaking with?

b. 香港 位于 那里？
      Hong Kong place at where?
      Where is Hong Kong located?

In the PP complement configuration, composition (B) allows an analysis consistent with object *wh*-extraction:
Chapter 3. The analysis of special syntax

(72) Where is Hong Kong located?

Hong Kong 位于 哪里

NP (S[cl]/NP) / PP PP/NP S[wq] / (S[cl]/NP)

S/(S\NP) (S[cl]/NP) / NP S[wq] / (S[cl]/NP)

$\Rightarrow^B$

S[cl]/NP $\Rightarrow^B$

S[wq]

The PP adjunct configuration requires the *wh*-word to take its preposition as an argument:

(73) Who are you talking to?

你 跟 谁 说话

NP (VP/VP)/NP (S[wq]/NP) / (S[cl]/NP) / ((VP/VP)/NP) S[cl]/NP

(S[wq]/NP) / (S[cl]/NP) / (S[wq]/NP)

$\Rightarrow^B$

S[wq]/NP $\Rightarrow^B$

S[wq]

Finally, the *wh*-word of manner 怎么样 how, and the *wh*-word of cause, 为什么 why, are interrogative pro-forms for an entire phrase of manner or cause respectively, leading to the analysis:

(74) Why are you going?

你 为什么 去

NP (S[wq]/NP) / (S[cl]/NP) S[cl]/NP

(S[wq]/NP) / (S[cl]/NP)

$\Rightarrow^B$

S[wq]/NP $\Rightarrow^B$

S[wq]

We now introduce a CCG analysis for a structural asymmetry between the *wh*-word of manner 怎么样 how and the *wh*-word of cause 为什么 why, by projecting an account from the generative literature into CCG.
The 为什么 why/怎么样 how asymmetry

The *why*-word of reason 为什么 why and the *how*-word of purpose 怎么样 how pattern similarly in simple sentences:

(75)  a. 她 为什么骂 我的弟弟 ？
     3SG why scold 1SG DE younger.brother ?
     Why is she scolding my younger brother?

    b. 你 怎么样写 小说 ？
     2SG how write novel ?
     How do you write a novel?
(both from Li and Thompson (1989))

(75) suggests that because both *why*-words occur in VP pre-modifier position, both should receive ([S[wh] \ NP]/[S[del] \ NP], a functor from the predicate of a declarative sentence to the predicate of a *why*-word interrogative sentence.

However, while 为什么 why can take sentence-initial position, 怎么样 how cannot: 11

(76)  a. 为什么她 骂 我的弟弟 ？
     why 3SG scold 1SG DE younger.brother ?
     Why is she scolding my younger brother?

    b. *怎么样你 写 小说 ？
     how 2SG write novel ?
     *How do you write a novel?

A curious interaction with modal verbs was also observed by Lin (1992), where 为什么 why is ungrammatical when it follows the modal, but 怎么样 how is ungrammatical when it precedes the modal.

(77)  a. 她 为什么 应该 骂 我的弟弟 ？
     3SG why should scold 1SG DE younger.brother ?

    b. *她 应该 为什么 骂 我的弟弟 ？
     3SG should why scold 1SG DE younger.brother ?
     Why should she scold my younger brother?

(78)  a. *你 怎么样 应该 写 小说 ？
     2SG how should write novel ?

11 Li and Thompson (1989) observe that 怎么 zemme alone, as opposed to 怎么样 zemneyang can have the meaning *why* and hence occur sentence-initially. We consider 怎么 zemme (the *why*-word of reason why) a distinct morpheme to its occurrence as a *why*-word of purpose.
Chapter 3. The analysis of special syntax

b. 应该 怎么样 写 小说？
2sg should how write novel?
How should you write a novel?

Lin (1992) explains the asymmetry in terms of GB. According to Lin, 为什么 why is at Spec of CP and undergoes movement to a pre-VP position, explaining why it cannot follow a modal verb. On the other hand, 怎么样 how is immediately dominated by VP, preventing it from preceding a modal verb at I.

We develop a CCG solution to the asymmetry in the spirit of this analysis. According to Lin, 为什么 why is base-generated at Spec of CP, which corresponds to 为什么 why ⊨ S/S, while 怎么样 how is attached at VP, which corresponds to 怎么样 how ⊨ (S[cl]NP)/(S[cl]NP).

(79) a. 为什么 S/S [她 骂 我 的 弟弟]S
why 3sg scold 1sg de younger.brother
Why does she scold my younger brother?

b. 你 [怎么样 VP/VP [写 小说] VP]
2sg how write novel
How do you write a novel?

However, the category assigned to 怎么样 how prevents us from blocking *怎么样 应该 写 how should write while permitting 应该 怎么样 写 should how write, because the category of 应该 写 should write is S[cl]NP, the same category as 写 write alone.

We stipulate that modals are functors from unsaturated to saturated VP, where ‘saturation’ refers to whether the VP has been closed off, simulating the GB constraint that a modal at I cannot appear inside the VP.

应该 should ⊨ (S[dd]NP)/(S[dd]NP)

This allows the following category assignments to the two wh-words:

怎么样 how ⊨ VP[q]/VP[cl]

为什么 why ⊨ S[q]/S[cl]

which account for the canonical word order for why/how wh-word sentences:
3.7. wh-questions

(80)  

a. In what way do you leave?

$\begin{array}{c}
\text{你 } \text{怎么 } \text{走} \\
\text{2SG } \text{how } \text{go}
\end{array}$

$\begin{array}{c}
\text{VP} \left[ \frac{q_{sat}}{dcl} \right] / \text{VP} \left[ dcl \right] \\
\text{VP} \left[ \frac{q_{sat}}{dcl} \right] >
\end{array}$

$\begin{array}{c}
\text{S} \left[ \frac{q_{sat}}{dcl} \right] \\
\text{VP} \left[ \frac{q_{sat}}{dcl} \right] <
\end{array}$

b. Why do you leave?

$\begin{array}{c}
\text{为什么 } \text{你 } \text{走} \\
\text{why } \text{2SG } \text{go}
\end{array}$

$\begin{array}{c}
\text{S} \left[ \frac{q_{sat}}{dcl} \right] / \text{S} \left[ dcl \right] \times \text{NP} \\
\text{S} \left[ dcl \right] \times \text{NP} >
\end{array}$

$\begin{array}{c}
\text{S} \left[ \frac{q_{sat}}{dcl} \right] \times \text{NP} \\
\text{S} \left[ \frac{q_{sat}}{dcl} \right] <
\end{array}$

The VP-initial position of 为什么 why in (75-a) is derived using forward crossed composition:

(81)  

Why do you leave?

$\begin{array}{c}
\text{你 } \text{为什么 } \text{走} \\
\text{2SG } \text{why } \text{go}
\end{array}$

$\begin{array}{c}
\text{NP} \text{S} \left[ \frac{q_{sat}}{dcl} \right] / \text{S} \left[ dcl \right] \times \text{NP} \\
\text{S} \left[ dcl \right] \times \text{NP} > \text{B}_x
\end{array}$

$\begin{array}{c}
\text{S} \left[ \frac{q_{sat}}{dcl} \right] \times \text{NP} \\
\text{S} \left[ \frac{q_{sat}}{dcl} \right] < \text{B}_x
\end{array}$

Finally, the $\pm sat$ feature accounts for the asymmetric interaction between 为什么 why/怎么样 how and modal verbs:

(82)  

a. Why should you leave?

$\begin{array}{c}
\text{你 } \text{为什么 } \text{应该 } \text{走} \\
\text{2SG } \text{why } \text{should } \text{go}
\end{array}$

$\begin{array}{c}
\text{NP} \text{S} \left[ \frac{q_{sat}}{dcl} \right] / \text{S} \left[ dcl \right] \times \text{VP} \left[ \frac{dcl}{q_{sat}} \right] / \text{VP} \left[ \frac{dcl}{sat} \right] \text{VP} \left[ dcl \right] >
\end{array}$

$\begin{array}{c}
\text{VP} \left[ \frac{dcl}{sat} \right] \\
\text{VP} \left[ \frac{q_{sat}}{dcl} \right] < \text{B}_x
\end{array}$

$\begin{array}{c}
\text{VP} \left[ \frac{q_{sat}}{dcl} \right] \\
\text{S} \left[ \frac{q_{sat}}{dcl} \right] <
\end{array}$
b. *(intended) Why should you leave?

```
你 应该 为什么 走
```

```
*2SG should why go
```

```
NP VP[+dcl]/VP[-dcl] S[q]/S[+dcl] VP[+dcl] VP[q]
```

(83) a. *(intended) How should you leave?

```
你 怎么样 应该 走
```

```
*2SG how should go
```

```
NP VP[q]/VP[+dcl] VP[q]/VP[-dcl] VP[+dcl] VP[q]
```

b. How should you leave?

```
你 应该 怎么样 走
```

```
2SG should how go
```

```
NP VP[+q]/VP[-q] VP[q]/VP[-dcl] VP[+dcl] VP[q]
```

```
S[q]
```

CCG analyses of *wh*-movement languages such as English showed that *wh*-movement syntax can be captured effectively in CCG without proposing movement or transformations (Hockenmaier, 2003). We have supplemented the literature with an account of *wh*-in-situ, demonstrating that this syntax also yields a succinct analysis in CCG. Finally, we have shown how a Chinese *wh*-word asymmetry from generative syntax can be captured in CCG through the feature mechanism.

### 3.8 Summary

We have presented a complete analysis of all of the NLD types covered by the Penn Chinese Treebank, demonstrating that CCG is capable of accounting for Chinese NLD-projecting constructions. We are now ready to describe Chinese CCGbank, the automatic corpus conversion from the Penn Chinese Treebank, showing that we can reshape its analyses to obtain a CCG corpus which embodies the analyses of this and the preceding chapter.
Then, having obtained a wide-coverage corpus of Chinese CCG derivations which incorporate our abstract analyses, we will pay particular attention in Chapter 7 to how well the NLDS arising from the analyses of this chapter can be retrieved in the practical, wide-coverage parsing setting, by performing error analyses on three CCG parsers trained on Chinese CCGbank.
Chapter 4

Creating Chinese CCGbank

The analysis of Chinese syntax in the previous two chapters can serve as manual annotation guidelines for the annotation of Chinese text with CCG structure. However, the alternative of corpus conversion has been shown to achieve high-quality corpora without the massive expense of manual annotation (Xia et al., 2000b; Cahill et al., 2002; Hockenmaier, 2003; Miyao et al., 2004). The resulting corpora are developed more quickly, largely retain compatibility with their source corpus, and greatly reduce the time burden and financial burden of manual annotation. Corpus conversion is an ideal example of reusing an existing resource to create new resources, and also directly enables the parser-aided corpus conversion methodology to be described in Chapter 7.

CCGbank was developed by Hockenmaier (2003) to demonstrate that corpus conversion could successfully extract a wide-coverage CCG grammar, and that the ability of CCG to transparently recover deep dependencies could lead to state-of-the-art parsing models. CCGbank also enabled the creation of the first wide-coverage CCG parsing models (Hockenmaier and Steedman, 2002b), as well as the state-of-the-art statistical CCG parser C&C (Clark and Curran, 2007b).

The Penn Chinese Treebank (Xue et al., 2005) fills in Chinese the gap filled by the landmark Penn Treebank in English, by creating a corpus which is both similar to, and improves upon, the Penn Treebank style of annotation. One of the specific goals of the Penn Chinese Treebank annotation was to improve the reliability of corpus conversions such as the one in this work, relative to its precursor the Penn Treebank.
This chapter traces the development of Penn Chinese Treebank and CCGbank, the two corpora most relevant to this work, and their annotation characteristics which inform the development of Chinese CCGbank. Then, we present the conversion algorithm which imposes the analysis of Chinese syntax from the previous two chapters on Penn Chinese Treebank trees in order to produce Chinese CCGbank.

### 4.1 The Penn Chinese Treebank

The Penn Chinese Treebank (PCTB; Xue et al., 2005) is the largest corpus of syntactically annotated Chinese text available. The latest version, PCTB 7.0, includes newswire text, magazine articles, and transcribed speech, reflecting publications and media from across the Sinosphere. The PCTB has been the primary focus of the English-language literature in Chinese parsing research, beginning with Bikel and Chiang (2000) who trained a parser on version 2.0 of the Treebank.

Penn Chinese Treebank 6.0, which we use for all experiments in this dissertation, augments the corpus with a section consisting of transcribed broadcast news from the ACE shared task. Because of the change in text type, some systems exclude the broadcast news sections (Harper and Huang, 2009; Guo et al., 2011). Penn Chinese Treebank 7.0 added a large quantity of text originating from Chinese blogs and newsgroups, and a broadcast conversation section, consisting of transcribed television interviews. The latter supplement motivated the addition of new phrase-level tags which handle incomplete constituents resulting from dysfluencies.

Table 4.1 compares the basic statistics of three versions of the Penn Chinese Treebank against the Penn Treebank, demonstrating its relative size and growth during development. The differences in aggregate statistics between versions 6.0 and 7.0 of the Penn Chinese Treebank reflect the text type of the new additions. While the average sentence length of versions 5.0 and 6.0, which are mostly composed of journalistic text, exceeds that of the PTB, the corresponding statistic on version 7.0 is more in line with that of the PTB. However, traces are consistently more frequent in the PCTB annotation, in line with the observation of Guo et al. (2007b) that Chinese syntax generates NLDs more frequently than English.

Table 4.2 reinforces the argument that particular NLD types occur more frequently in Chinese; 41% of PCTB sentences contain an extraction instance, compared to only 10% of English sentences, and a large proportion of sentences involve loose coordination, defined as full clauses separated only by commas. Loose coordination can lead to considerable parser ambiguity, because in addition to its role as a conjunction word, it also plays the
Figure 4.1: A Penn Chinese Treebank derivation (glosses provided by us)
Chapter 4. Creating Chinese CCGbank

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Size in</th>
<th>Average per sent.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sentences</td>
<td>words</td>
</tr>
<tr>
<td>Penn</td>
<td>5.0</td>
<td>18,804</td>
</tr>
<tr>
<td>Chinese</td>
<td>6.0</td>
<td>28,295</td>
</tr>
<tr>
<td>Treebank</td>
<td>7.0</td>
<td>51,451</td>
</tr>
<tr>
<td>Penn Treebank</td>
<td>3.0</td>
<td>49,208</td>
</tr>
</tbody>
</table>

Table 4.1: Comparison of corpora

<table>
<thead>
<tr>
<th>Corpus</th>
<th>% of sentences with</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>relative clauses</td>
</tr>
<tr>
<td>Penn Chinese Treebank</td>
<td>6.0</td>
</tr>
<tr>
<td>Penn Treebank</td>
<td>3.0</td>
</tr>
</tbody>
</table>

Table 4.2: Comparison of structural characteristics

Figure 4.2: Composition of PCTB 6.0

role of prosodic marker or disambiguator, causing a large amount of parser ambiguity compounded by the length of the sentence.

The PTB (Marcus et al., 1994) annotation style poses problems for some NLP applications. The structure of PTB productions is largely unconstrained, leading to a large number of production rules in a grammar induced from it. Head structure is also not marked in the PTB, and head-finding heuristics (e.g. those in Magerman (1994)) are typically used by applications which require identification of phrasal heads, including constituent-to-dependency conversion, feature extraction in parsers, and corpus conversions.

The complement/adjunct distinction is also underspecified in PPs, motivating the use of heuristics to recover the distinction between the two kinds of PPs below:
4.1. The Penn Chinese Treebank

(1)  a. I built a bunny [ for John’s birthday. ] PP-Adjunct
    b. I asked [ for a clarification. ] PP-Complement

Other potential sources of annotation noise include the absence of NP internal structure (since addressed by Vadas and Curran (2008)), and an inconsistent complement/adjunct distinction. As such, researchers implementing corpus conversions from the Penn Treebank have unavoidably introduced sources of error, arising from underspecified representation in many aspects of the PTB annotation scheme, particularly in applications requiring accurate head structure annotations.

Despite its impoverished representation, the influence of the Penn Treebank as the core resource in English-language NLP is unquestionably immense. It has long been the standard resource in parsing (Magerman, 1995; Collins, 1999; Charniak, 2000), POS tagging (Ratnaparkhi, 1996; Brants, 2000) and NP chunking (Tjong Kim Sang and Buchholz, 2000). While the Penn Chinese Treebank is a much newer corpus, it has assumed a similar position in the Chinese NLP literature, as a prominent resource for parsing (Bikel and Chiang, 2000; Levy and Manning, 2003; Duan et al., 2007; Li et al., 2011; Zhang and Clark, 2011) and word segmentation (Xue, 2003; Jiang et al., 2009).

Xue et al. (2005), recognising the power of corpus conversion, reports that one of the design decisions of the Penn Chinese Treebank was to eliminate some of these sources of error by adhering to a more principled style of annotation compared to the Penn Treebank. Figure 4.1 is a PCTB tree illustrating some aspects of its annotation. While the POS tag scheme, the use of functional tags (e.g. NP-SBJ), and the analysis of NLDs through traces (*T*) are reminiscent of the PTB annotation, PCTB trees encode gold-standard head structure, and the production rules are highly constrained to reduce sparsity.

We focus on the characteristics of the Penn Chinese Treebank which enable us to extract a high-quality Chinese CCGbank.

4.1.1 Differences between PTB and PCTB

The PCTB tagset

Table 4.3 reproduces the PCTB tagset, with examples of each tag. Certain differences from the Penn Treebank tagset arise from typological differences between the two languages. For instance, while the PTB tagset distinguishes number on nouns (NNS) and morphological forms of verbs (VBD), the PCTB tagset does neither.
### Chapter 4. Creating Chinese CCGbank

<table>
<thead>
<tr>
<th>Tag</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>VA</td>
<td>Adjectival verb</td>
</tr>
<tr>
<td>VC</td>
<td>Copula</td>
</tr>
<tr>
<td>VE</td>
<td>Existential verb</td>
</tr>
<tr>
<td>VV</td>
<td>Verb (other)</td>
</tr>
<tr>
<td>NR</td>
<td>Proper noun</td>
</tr>
<tr>
<td>NT</td>
<td>Temporal noun</td>
</tr>
<tr>
<td>NN</td>
<td>Common noun</td>
</tr>
<tr>
<td>PN</td>
<td>Pronoun</td>
</tr>
<tr>
<td>LC</td>
<td>Localiser</td>
</tr>
<tr>
<td>DT</td>
<td>Determiner</td>
</tr>
<tr>
<td>CD</td>
<td>Cardinal number</td>
</tr>
<tr>
<td>OD</td>
<td>Ordinal number</td>
</tr>
<tr>
<td>M</td>
<td>Measure word</td>
</tr>
<tr>
<td>AD</td>
<td>Adverb</td>
</tr>
<tr>
<td>P</td>
<td>Preposition</td>
</tr>
<tr>
<td>CC</td>
<td>Coordinating conjunction</td>
</tr>
<tr>
<td>CS</td>
<td>Subordinating conjunction</td>
</tr>
<tr>
<td>SP</td>
<td>Sentence-final particle</td>
</tr>
<tr>
<td>AS</td>
<td>Aspect marker</td>
</tr>
<tr>
<td>ETC</td>
<td>etc.</td>
</tr>
<tr>
<td>MSP</td>
<td>Miscellaneous particle</td>
</tr>
<tr>
<td>JJ</td>
<td>Adjective</td>
</tr>
<tr>
<td>PU</td>
<td>Punctuation</td>
</tr>
<tr>
<td>FW</td>
<td>Foreign word</td>
</tr>
<tr>
<td>IJ</td>
<td>Interjection</td>
</tr>
<tr>
<td>ON</td>
<td>Onomatopoeia</td>
</tr>
<tr>
<td>DER</td>
<td>Adverb marker 地 de</td>
</tr>
<tr>
<td>DEV</td>
<td>Particle 得 de</td>
</tr>
<tr>
<td>DEC</td>
<td>Relativiser</td>
</tr>
<tr>
<td>DEG</td>
<td>Generalised modifier</td>
</tr>
<tr>
<td>LB</td>
<td>Long 被 bei</td>
</tr>
<tr>
<td>SB</td>
<td>Short 被 bei</td>
</tr>
<tr>
<td>BA</td>
<td>把 ba</td>
</tr>
</tbody>
</table>

Table 4.3: The Penn Chinese Treebank word-level tagset
### Table 4.4: The Penn Chinese Treebank phrase-level tagset

<table>
<thead>
<tr>
<th>Tag</th>
<th>Head</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>-</td>
<td>张三打了李四 Zhangsan hit Lisi.</td>
</tr>
<tr>
<td>VP</td>
<td>V*</td>
<td>被张三打了 got hit by Zhangsan</td>
</tr>
<tr>
<td>LCP</td>
<td>LC</td>
<td>桌子上 on the table/the top of the table</td>
</tr>
<tr>
<td>NP</td>
<td>N*</td>
<td>欧盟欧洲委员会驻华代表团 European Council delegation to China</td>
</tr>
<tr>
<td>PP</td>
<td>P</td>
<td>在桌子上 on the table</td>
</tr>
<tr>
<td>QP</td>
<td>CD</td>
<td>一百四十八项 148 items</td>
</tr>
<tr>
<td>ADJP</td>
<td>JJ</td>
<td>大型、特大型 large and extralarge</td>
</tr>
<tr>
<td>ADVP</td>
<td>AD</td>
<td>也可能 also possibly</td>
</tr>
<tr>
<td>CLP</td>
<td>M</td>
<td>整个 the entire ...</td>
</tr>
<tr>
<td>CP</td>
<td>SP</td>
<td>DEC</td>
</tr>
<tr>
<td>DP</td>
<td>DT</td>
<td>各类 each type</td>
</tr>
<tr>
<td>DNP</td>
<td>DEG</td>
<td>中国银行的～ ... of the Bank of China</td>
</tr>
<tr>
<td>DVP</td>
<td>DEV</td>
<td>～得快 do ...fast</td>
</tr>
<tr>
<td>UCP</td>
<td>-</td>
<td>距离近、方便 distance is close, and convenient</td>
</tr>
<tr>
<td>PRN</td>
<td>-</td>
<td>（包括加工丝） (including processed silk)</td>
</tr>
<tr>
<td>FRAG</td>
<td>-</td>
<td>在国际新闻方面。 As for international news.</td>
</tr>
<tr>
<td>LST</td>
<td>-</td>
<td>第二： Second:</td>
</tr>
</tbody>
</table>
On the other hand, the PTB tagset distinguishes multiple senses of some overloaded function words through their pos tags: the particle 的 de (DEC, DEG, DER) and the two frames of the passiviser 被 bei (LB, SB). Certain other tags are simply renamed from their PTB equivalents — adverbs (AD), prepositions (P) and pronouns (PN).

Similarly, the phrase-level tagset (Table 4.4) which enumerates the tags that occur at internal nodes, shares many categories with the PTB tagset and traditional generative grammar (NP, VP, PP), but adds tags for phrases projected by heads particular to Chinese such as localisers (LCP) and measure words (CLP).

The tag IP for inflectional phrase is identical to the PTB tag S; the name reflects a theory of generative grammar which suggests that the sentence is a phrasal projection of an Infl (inflection) head (Haegemann, 1994), even though no such head exists in the PTB annotation.

The tag LCP is the projection of a localiser head, and occurs in a distribution similar to the English PP. In the below example, while the English preposition on carries the semantic content, the corresponding construction in Chinese has the dummy preposition 在 zai, with the semantic content carried by the localiser 上 on:LC instead.

(2) 在 桌子 上
    at table on:LC
    on the table

The tags CLP (classifier phrase) and QP (quantifier phrase) occur in the annotation of NPs modified by a quantity. Recall from Section 2.4.1 that any quantification of an NP must involve a measure word, which comes between the quantity and the NP. The PTB captures this syntax by having the numeral (CD) take the CLP as complement, projecting a QP (quantifier phrase), which then modifies the NP.

(3) a few hundred construction companies

\[
\begin{array}{c}
\text{NP} \\
\text{QP} \\
\text{CD} \quad \text{CLP} \quad \text{NN} \quad \text{NN} \\
\text{数} \quad \text{M} \quad \text{建筑} \quad \text{公司} \\
\text{few.hundred} \quad \text{construction} \quad \text{company} \\
\end{array}
\]
4.1. *The Penn Chinese Treebank*

Sentences with sentence-final particles, such as the question particle ㄜ ma and 因 ne, are analysed as being headed by the particle, which projects a complementiser head (CP). The relativiser ㄜ de is also treated as a head of CP.

**Phrase- and word-level tags**

Penn Treebank trees often underspecify NP internal structure, which leads to a large number of unfactored productions for NPs. In the below tree, the head noun exposures is siblings with an adverb and an adjective:

(4)  
```
  VP
  └── VBP prevent
      └── NP
          └── RB even JJ brief NNS exposures
```

A problem with this representation is that each combination of modifiers results in a distinct production rule, leading to a proliferation of production rules:

(5)  
```
  VP
  └── VBP prevent
      └── NP
          └── RB even RB just RB very JJ brief NNS exposures PP
                          └── IN to NN ash
```

The factoring of production rules according to the head structure of the phrase was an important contribution of the Collins (1999) model, and effectively allows these sparse production rules to be factored into left- or right-branching structures, enabling the estimation of rule probabilities despite the sparseness of production rules.

The Penn Chinese Treebank controls the sparseness of production rules in the annotation by constraining their structure. The first constraint is the distinction between phrase- and word-level tags — phrase-level tags are constrained to occur at a derivation’s internal nodes, and word-level tags at the leaves. While this distinction exists in the Penn Treebank, which also distinguishes POS tags from syntactic tags (Marcus et al., 1994), both word-level and phrase-level tags can freely occur on the right hand side of a production rule.

By constraining the structure of production rules, with respect to the position of word- and phrase-level tags occurring on the right hand side of a rule, the Penn Chinese Treebank both encodes gold-standard complement/adjunct distinctions, and factors produc-
tion rules to mitigate the impact of sparsity. The next section describes the structure of these constraints on production rules.

Constraints on productions

The central annotation difference separating PTB and PCTB is that the Chinese corpus applies stringent constraints on the structure of its production rules. As introduced in the previous section, the Penn Chinese Treebank distinguishes phrase-level categories (NP, ADVP) from lexical categories (NN, JJ and so on).

Consider the below PCTB-style structure, which corresponds to (4).

\[
(6)
\begin{array}{c}
\text{VP} \\
\quad \text{VBP prevent} \\
\quad \quad \text{NP} \\
\quad \quad \quad \text{ADVP} \\
\quad \quad \quad \quad \text{RB even} \\
\quad \quad \quad \quad \quad \text{ADJP} \\
\quad \quad \quad \quad \quad \quad \text{JJ brief} \\
\quad \quad \quad \quad \quad \quad \quad \text{NNS exposures}
\end{array}
\]

In (4), the head noun and its modifiers are siblings under NP, and the internal structure of the NP is left underspecified. The PCTB-style bracketing makes the modification structure explicit, and the single production \( NP \rightarrow \text{RB JJ NNS} \) is decomposed into two more general productions \( NP \rightarrow \text{ADVP NP} \) and \( NP \rightarrow \text{ADJP NP} \), leading to fewer and more robust production rules.

Additionally, the head structure of a node's children is encoded in the configuration of the parent tag and its child tags.\(^1\) For instance, a production rule in which the first child has a word-level tag, and the other children are all phrase-level tags \( \text{VV NP PP} \rightarrow \text{VP} \) encodes head-initial complementation:

\[
(7)
\begin{array}{c}
\text{VP} \\
\quad \text{VV gives} \\
\quad \quad \text{NP} \\
\quad \quad \quad \text{PP} \\
\quad \quad \quad \quad \text{DT the} \\
\quad \quad \quad \quad \quad \text{NN pumpkin} \\
\quad \quad \quad \quad \quad \quad \text{IN to} \\
\quad \quad \quad \quad \quad \quad \quad \text{NNP John}
\end{array}
\]

\(^1\) The Collins (1999) head finding rules recover an approximation to head structure, while the PCTB annotation encodes it directly.
Similarly, a production whose right hand side consists only of phrase-level tags, for instance \( \text{ADVP} \rightarrow \text{VP} \), indicates the adjunction relation:

\[
\begin{array}{c}
\text{ADVP} \quad \text{VP} \\
\mid \quad \mid \\
\text{ADVP} \quad \text{VBD devoured} \quad \text{NP} \\
\mid \quad \mid \\
\text{RB very} \quad \text{RB quickly} \quad \text{DT the} \quad \text{NP} \\
\mid \quad \mid \\
\quad \quad \text{NN pumpkin} \\
\end{array}
\]

A consequence of these constraints is the prevalence of unary projections from word- to phrase-level tags in the annotation, which are necessary to ensure the constraints on production rules. For instance, (9-a) is an ill-formed production, because the two word-level categories \( \text{VV} \) and \( \text{NN} \) cannot be siblings. The addition of a unary projection \( \text{NN} \rightarrow \text{NP} \) fixes the derivation, with the production rule \( \text{VV NP} \rightarrow \text{VP} \) correctly indicating head-initial complementation.

\[
\begin{array}{c}
\text{VP} \\
\mid \\
\text{VV ate} \quad \text{NN pumpkins} \\
\mid \\
\text{VP} \\
\mid \\
\text{VV ate} \quad \text{NP} \\
\mid \\
\text{NN pumpkins} \\
\end{array}
\]

The full set of constraints on \( \text{PCB} \) production rules, and the headedness relation which each configuration indicates, is given in Table 4.5.

Distinguishing complementation and adjunction greatly simplifies the recovery of \( \text{CCG} \) category structure later in the conversion process.

**Composition**

The Penn Treebank is relatively homogeneous owing to its domain — U.S. English text from the Wall Street Journal from a contiguous time period (1987-9) — and owing to the short but concentrated duration of its annotation effort.\(^2\) By contrast, the Penn Chinese

\(^2\) Marcus (2011) is a retrospective of the Penn Treebank's development.
Chapter 4. Creating Chinese CCGbank

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Head structure</th>
<th>Headedness</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP → XP-SBJ VP</td>
<td>Predication</td>
<td>Right</td>
</tr>
<tr>
<td>XP → XP [(PU</td>
<td>CC) XP]^+</td>
<td>Coordination</td>
</tr>
<tr>
<td>HP → H (YP ...)</td>
<td>Head-initial complementation</td>
<td>Left</td>
</tr>
<tr>
<td>HP → (YP ...) H</td>
<td>Head-final complementation</td>
<td>Right</td>
</tr>
<tr>
<td>XP → YP XP</td>
<td>Left-adjunction</td>
<td>Right</td>
</tr>
<tr>
<td>XP → XP YP</td>
<td>Right-adjunction</td>
<td>Left</td>
</tr>
</tbody>
</table>

Table 4.5: Configurations of rules, with word-level tag H and phrase-level tags XP, YP, HP

Treebank 7.0 data encompasses newswire from five news agencies, transcribed broadcast news and speech from ten providers, magazine articles, and a selection of web and newsgroup text, as a result of its growth over twelve years, from the original release in 2000 through to the most recent release in 2012. Therefore, when considered as a single corpus, PCTB has an uneven character.

4.1.2 Dataset splits

We are aware of five splits of the Penn Chinese Treebank used in the literature. One is the round-robin split of Huang et al. (2007), which partitions every block of ten files in the treebank into one development file, one test file and eight training files. This split is applied to PCTB 6.0 and is followed by Harper and Huang (2009), Wang (2008), Huang and Harper (2009) and Wang and Zong (2010).

A second split is the most common in the literature, but only applies to PCTB 2.0. This particular split originates from Bikel and Chiang (2000), the seminal work in parsing with the Penn Chinese Treebank, and divides the corpus into training, development and test sections formed from contiguous tracts of the treebank. This split is used by work including Chiang and Bikel (2002), Bikel (2004), Xiong et al. (2005) and Wang et al. (2006).

A third split is defined by Duan et al. (2007) over PCTB 5.0 and used by Zhang and Clark (2008), Huang and Sagae (2010), Chen et al. (2009), Zhang and Clark (2009), Li et al. (2011), Ponvert et al. (2011) and many others.

Levy and Manning (2003) observed that the data split of Bikel and Chiang (2000) leads to high lexical sparsity, as its development set happens to consist of non-economic topics. In particular, they observe that the fairly common lexical item 分 fen is never attested as a
measure word in the development set. They use a fourth split which redistributes the split between the training and development sets while maintaining contiguity.

Finally, a fifth split is suggested by the file manifest of the Penn Chinese Treebank 6.0 distribution. However, this split, which is not consistent with any of the above splits, is used by relatively few systems, including (Guo et al., 2011).

Accordingly, we are faced with a choice between a number of splits, some of which are not defined over PCTB 6.0, and some of which are obsolete. Because the split deriving from Duan et al. (2007) is the most common and is defined over PCTB 5.0, we use a split which is consistent with this split, adding data from the new data in PCTB 6.0, while maintaining the same proportions. This leads to the split in Table 4.6, which we adopt as the suggested partition for Chinese CCGbank, and the partition used in all our experiments described in Chapter 6.

<table>
<thead>
<tr>
<th></th>
<th>PCTB 5</th>
<th>+PCTB 6</th>
<th>#sents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>1–815, 1001–1136</td>
<td>2000–2980</td>
<td>22033</td>
</tr>
<tr>
<td>Test</td>
<td>816–885, 1137–1147</td>
<td>3030–3145</td>
<td>2758</td>
</tr>
<tr>
<td>Dev</td>
<td>900–931, 1148–1151</td>
<td>2981–3029</td>
<td>1101</td>
</tr>
</tbody>
</table>

Table 4.6: PCTB 5 and 6 dev/train/test splits

4.2 CCGbank

While categorial grammars and CCG had previously been applied to the analysis of language fragments only, the development of CCGbank (Hockenmaier, 2003) enabled its representational benefits to be tested in the wide-coverage parsing setting for the first time.

One strong motivation for the development of CCGbank was to test whether a formalism offering a consistent account of both local and non-local dependencies would outperform one which does not recover non-local dependencies.

To test this hypothesis, Hockenmaier (2003) developed a Collins (1999)-style generative model over CCG normal-form derivations, achieving results in unlabelled dependency recovery which were competitive with the contemporary state of the art.

As the first wide-coverage CCG corpus in the literature, the annotation conventions of CCGbank have influenced the development of subsequent CCG corpora. An important
Chapter 4. Creating Chinese CCGbank

Characteristic of CCGbank-style grammars is the use of type-change rules, which Hockenmaier and Steedman (2002a) argue is critical to control modifier proliferation, the feature mechanism (Section 1.5.1) to encode morphological distinctions, the absorption punctuation analysis, and the use of the two-step coordination rule (Rule 4) instead of lexicalised coordination or the syncategorematic coordination of Steedman (2000).

Because of the methodological parallels with our work, which also extracts a CCGbank-style grammar from a Penn-style treebank, the Chinese CCGbank annotation adopts aspects of the English CCGbank annotation.

4.3 Comparisons with other conversion algorithms

4.3.1 English CCGbank

Hockenmaier (2003) describes the English CCGbank conversion algorithm as a pipeline of three stages: determining constituent type, binarisation, and category assignment. These subroutines correspond to the stages mark, binarise and label in our system. Hockenmaier (2003) uses head-finding rules adapted from Collins (1999) and Magerman (1994), which identify as the head the leftmost or rightmost child (depending on the parent's non-terminal label) which matches against a list of candidate head labels. We showed in this chapter that the head/complement/adjunct annotations encoded in the configuration of PCTB nodes is sufficient to induce gold-standard head structure for our corpus conversion.

4.3.2 CCGbanks from dependency corpora: TUT-CCG Italian CCGbank and a German CCGbank

Automatic corpus conversions exist in the CCG literature from both constituent treebanks and dependency treebanks. The conversion by Hockenmaier (2006) of the German dependency corpus TIGER, and the conversion by Bos et al. (2009) of part of the Italian TUT corpus into CCG derivations are the two examples of conversions from dependency corpora in the CCG literature.

Both conversion algorithms involve an additional conversion stage in which the dependency graphs are first converted into trees according to the annotated head structure. The remainder of the TIGER and TUT conversions consists of binarisation and category assignment, as in the conversion from constituent treebanks.
TIGER does not mark the traces resulting from extraction, so Hockenmaier (2006) detects instances of extraction configurationally, finds the gapped argument, and percolates the “trace” down the category structure of the tree. The process which the TUT conversion uses to recover constituent structure from dependency graphs inserts Penn-style traces, allowing Bos et al. (2009) to recover extraction NLDS the same way as in conversions from constituent treebanks.

4.4 Summary of the conversion algorithm

The Chinese CCGbank corpus conversion is a procedure which takes as input a PCtb tree, and produces a normal-form ccg derivation which preserves the dependencies inherent in the PCtb analysis. We achieve this by passing the PCtb tree through a pipeline which incrementally transforms it into a normal-form ccg derivation. This system architecture is inherited from the English CCGbank conversion algorithm (Hockenmaier, 2003), who used a pipeline of conversion stages to transform Penn Treebank trees into ccg derivations. Because the Penn Chinese Treebank annotation style was consciously modelled after the Penn Treebank style (Xue et al., 2005), the Chinese CCGbank conversion algorithm too is modelled after the English CCGbank conversion algorithm.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>Pre-process trees.</td>
</tr>
<tr>
<td>Mark</td>
<td>Explicitly annotate heads, arguments and constructions.</td>
</tr>
<tr>
<td>Binarise</td>
<td>Binarise trees according to head structure.</td>
</tr>
<tr>
<td>Label</td>
<td>Re-label nodes with categories.</td>
</tr>
<tr>
<td>Fix traces</td>
<td>Create trace-free analyses.</td>
</tr>
<tr>
<td>Fix modifiers</td>
<td>Generalise modifier categories.</td>
</tr>
</tbody>
</table>

Table 4.7: Stages of the generation algorithm

4.5 The generation algorithm

The five stages in our conversion algorithm in Table 4.7 — filtering and preprocessing, marker annotation, binarisation, category labelling, and trace elimination — each arise
from a point of difference between the PCTB style of annotation, and a CCGbank style grammar. We will justify the role and effect of each stage in turn.

**Filtering**

We perform pre-processing to modify the PCTB tokenisation, splitting or joining tokens from the original annotation. Because we want all processing stages to operate on the modified tokenisation, changes to the tokenisation occur first.

**Marking**

Penn Chinese Treebank trees encode gold-standard complement/adjunct distinctions, and distinguish heads from arguments. This is a departure from the Penn Treebank annotation style, which did not reliably distinguish PP complements from adjuncts, and does not indicate syntactic heads explicitly:

```
(10)
S
  NP-SBJ
    NNS prices
  VP
    VV rose
    PP
      according to figures
```

Hockenmaier (2003) developed heuristics to recover the PP complement/adjunct distinction, and use the Magerman (1994) head-finding rules to recover head structure, unavoidably introducing a source of error.

The PCTB annotation, on the other hand, does encode gold-standard head/complement/adjunct (HCA) distinctions. However, the head information is implicit in the structure of a given production, rather than an explicit annotation on the labels on either side of a production rule. This means that when that structure is transformed, as will be necessary to enforce CCG’s binary-branching requirement, and to re-analyse certain constructions, these distinctions become lost.

The marking phase addresses this problem by explicitly encoding HCA distinctions into the node labels, which enables their information to be preserved when the tree is restructured. For instance, the implicit HCA relations present in Figure 4.3 are represented explicitly as tags (e.g. :a, :h, :l) on node labels in the corresponding marked output (Figure 4.4).
4.5. The generation algorithm

Binarisation

In the PCTB annotation style, a head and all its complements are represented as siblings, as are the phrasal projection of a head and its modifier phrases.

Because CCG combinatory rules accept at most two input categories, all $k$-way branching structures must be binarised in accordance with the head structure made explicit in the previous phase. For instance, in Figure 4.6a, the right-binarisation leads to incorrect head structure for head-initial verb complementation; the correct binarisation in Figure 4.6b correctly has the verb collect its arguments one by one.

Category labelling

The binarised structures are then labelled recursively top-down with CCG categories in accordance with the marked head structure. This yields trees whose nodes are labelled with CCG categories, but may still contain traces.
Chapter 4. Creating Chinese CCGbank

Figure 4.6: Binarisation must reflect head structure

Trace elimination

PCCTB inherits an annotation style from the Penn Treebank in which constructions involving long-distance dependencies arise from movement from D-structure, leaving phonologically unrealised elements called traces in the derivation. Because CCG is a monostratal grammar formalism, and because the Principle of Adjacency in Steedman (2000) prohibits combinatorial rules from applying to phonologically unrealised elements, a CCGbank corpus conversion from such a source corpus must transform instances of these constructions into trace-free analyses which preserve the dependencies from the source tree.

4.6 The Chinese CCGbank conversion algorithm

This section presents the conversion algorithm, focusing on its implementation and the issues which shaped its construction. Algorithm 1 outlines the stages of the conversion algorithm, from preprocessing to the generation of gold-standard dependencies.

4.6.1 PREPROCESS($T$)

The preprocessing phase can modify the annotation in a way which affects all subsequent stages. Accordingly, annotation changes which affect the segmentation — merging or splitting tokens — are performed in preprocessing.
Algorithm 1 The Chinese CCGbank conversion pipeline

Precondition: a PCTB tree $T$

1. **PREPROCESS**($T$) ▷ Perform tokenisation changes and fixes
2. **MARK**($T$) ▷ Make head/complement/adjunct structure explicit
3. **BINARISE**($T$) ▷ Binarise according to head structure
4. **CATLAB**($T$) ▷ Label nodes with CCG categories
5. **FIX**($T$) ▷ Reshape syntax without traces
6. $T$.DEPS ← **MKDEPS**($T$) ▷ Induce dependencies

The preprocessing phase modifies the segmentation for three verb compounding strategies: verb coordination (**VCD**), the VNV construction (**VNV**) and the V-的 del/不 bu-V construction (**VPT**).

Algorithm 2 **PREPROCESS**($T$)

Precondition: a PCTB tree $T$

1. for node ∈ **NODES**($T$) do
2. if node.TAG ∈ {**VCD**, **VNV**, **VPT**} then
3. merge children of node into a single token with the tag of node
4. if node is a leaf and contains an interpunct (·) then
5. split node’s lexical item on the interpunct
6. repair projections or functional tags on node

Infixing morphology

Non-concatenative morphology is morphological marking which cannot be treated as the non-overlapping concatenation of roots and morphemes. Two constructions in Chinese involve non-concatenative morphology: the V-的 del/不 bu-V (verb potential) construction and the VNV construction. In the PCTB annotation, the infix in either case cleaves a single word into three tokens (Figure 4.7).

Both constructions are considered to be morphological operations in the literature (Packard, 2000), so we modify the annotation so that the result of these constructions is pre-composed. This is consistent with the treatment of morphology in the Penn Treebank and hence English CCGbank, where the lexicon treats lexical items (which may have underlying morphological structure) as pre-composed (Hockenmaier and Steedman, 2005).
Honnibal et al. (2010) subsequently experimented with applying morphological decomposition to English CCGBank, observing a modest improvement for parsing on gold standard POS tags, and a small improvement on automatically assigned POS tags. In their analysis, an inflected verb such as bought ⊢ \( S[pt]\NP \) would be decomposed into two morphemes — the root morpheme \textit{buy}- and a past participle morpheme \textit{-en}:

\[
\text{buy} \quad -en \\
\frac{S[b]\NP \quad (S[pt]\NP) \backslash (S[b]\NP)}{S[pt]\NP} \\
\ne
\]

We consider the implications of a similar representation of infixing morphology in Chinese CCGBank. Infixation can be treated categorically by extending the analysis of Honnibal et al. (2010) — just as the affix consumes the root to yield a new category, an infix may consume the two cleaved elements on either side of it, which we may notate \textit{LHS + Infix + RHS}:

\[
\text{LHS} \quad \text{Infix} \quad \text{RHS} \\
\frac{L \quad (O \backslash \wedge LR) \quad \wedge R \quad R}{O \backslash \wedge LR} \\
\quad > \quad O \quad \wedge Lorem
\]

In this analysis, the continuity of the root word is disturbed, so that the uncleaved root cannot be identified with the cleaved elements \((L, R)\). Furthermore, it requires the introduc-
tion of new atoms which correspond to cleaved left \((L)\) and right \((R)\) elements, aggravating the sparsity issues which the analogous analysis solves for English.\(^5\)

Another possibility is to consider affixation in the abstract, such that adfixation and infixation are treated uniformly:

\[
\begin{array}{c|c}
\text{Root} & \text{Affix} \\
\hline
R & R'\backslash R \\
\end{array}
\]

This maintains the continuity of the root word, without introducing new categories for the cleaved elements. However, such an analysis would posit a new level of representation between surface structure and phonetic form, and would require additional extra-lexical information to specify how the affixation is performed.

Our final choice of analysis is to render morphological structure invisible to CCG by only permitting morphologically composed words to enter into analyses. As mentioned previously, this is largely consistent with the (non-)analysis of morphology in English CCG-bank.\(^6\) This analysis also renders opaque the relationship between the canonical root word and the root word under infixation. However, it eliminates the need for the additional level of representation required in the abstract morphological analysis, or the unusual argument-taking behaviour of the morphologically decomposed representation.

In Chinese CCGbank, the fusion of the lexical items at the leaves of nodes with tag \texttt{VNV} or \texttt{VPT} is performed during the filtering stage of the conversion pipeline, ensuring that subsequent processing stages only see morphologically pre-composed lexical items.

**Bare verb coordination**

We fuse together the lexical items found under \texttt{VCD} nodes in the original annotation. This creates single tokens (e.g. 设计建设 design and build) consisting of coordinated bare verbs. This follows our analysis in Section 2.9.4.

---

\(^5\) The reason why the cleaved elements cannot be assigned full categories such as \texttt{S[dd]\NP} is that the residues of infixation do not necessarily constitute freestanding morphemes. This can readily be seen in Chinese, where, for instance, 得 \textit{de} in 得到 achieve is no longer a freestanding verb (with the meaning \textit{achieve}) in the contemporary language.

\(^6\) The genitive ’s is a clitic rather than an affix, because it can take phrasal scope.
Normalising foreign names

The Chinese National Standard on the use of punctuation marks\(^7\) states that the interpunct (间隔号 jiangehao; ·) should be used to separate the components of names transliterated from foreign languages or those of national minorities. Because the \texttt{pctb} tokenisation does not consider \(·\) to be a word separator, foreign names are represented as single tokens, possibly containing one or more interpuncts. To reduce the lexical sparsity arising from this tokenisation, we split foreign names on the interpunct, creating one leaf for each name component, including the interpunct:

\[
\begin{array}{c}
\text{NP-PN} \\
\text{NR 马拉克 Barack} \quad \text{PU ·} \quad \text{NR 欧巴马 Obama}
\end{array}
\]

Missing projections

The Chinese CCGbank conversion process relies on the configuration of a node and its child tags for the correct identification of head structure. Missing projections can cause one configuration to be identified as another, resulting in noisy CCG analyses.

For instance, the constraint that word-level categories should never be siblings is occasionally broken, resulting in an inability to identify the head category. We found 69 derivations (0.24\% of the corpus) which violate this constraint.

To maximise coverage, however, we implement a fallback rule in such cases which assumes right-headedness. We develop rules to address several cases in which this fallback rule leads to incorrect headedness. These cases are given in Appendix C.

Incorrect unary projections

In line with \(X'\) theory, phrases in \texttt{pctb} are intended to be endocentric — that is, phrases of type \(XP\) are projected by heads of type \(XP\). Some clear and systematic mis-taggings

---

\(^7\) \text{GB/T 15834-1995.}
indicate the wrong phrasal projection for a given head category, resulting in incorrect unary projections such as VV → NP or NN → CLP. We repair several such cases where it is clear that the annotation is incorrect, given in Appendix C. 119 (0.42%) of PCTB trees are affected by at least one change of this type.

**Summary of preprocessing changes**

Appendix C specifies all of the preprocessing changes we have made to the PCTB annotation in order to improve the coverage of the corpus conversion algorithm. This data is not only useful to implementors of corpus conversions from the PCTB, but for all researchers operating on the corpus, as it is the first account in the literature of representation issues which may be problematic for other NLP tasks based on the PCTB.

### 4.6.2 **MARK**(T)

Xue et al. (2005) reports that the choice to represent the complement/adjunct distinction explicitly in PCTB was intended to aid the development of transformed corpora such as Chinese CCGbank. We describe how the Chinese CCGbank conversion operates on the PCTB encoding of head structure described in Section 5.

The marking algorithm annotates the PCTB tag of each internal node with a *marker*, which preserves this headedness information, even after the nodes are restructured in the binarisation phase. Also, the algorithm identifies instances of Chinese syntax which require special head structure, including the Chinese CCGbank analyses of parenthetics (Section 2.8.3), topicalisation (Section 3.3) and argument cluster coordination (Section 2.6.4).

The marking algorithm is reproduced as Algorithm 3. The cascade of pattern matches from Line 5 to 18 are to be interpreted as follows. The tag and structure of node is compared against each configuration in order. If node’s configuration matches a particular case, apply the markers specified.

For instance, when **MARK** encounters the following sub-tree:

\[
\text{VP} \\
\text{VP} / \text{give} \quad \text{NP} / \text{他们} \quad \text{3PL} \quad \text{QP} / \text{3 dollars}
\]

it will try and match the tag structure against each pattern, failing until it reaches the head-initial complementation pattern in Line 12 of Algorithm 3. This pattern matches because
Chapter 4. Creating Chinese CCGbank

<table>
<thead>
<tr>
<th>Marker</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Adjunct</td>
</tr>
<tr>
<td>h</td>
<td>Head in complementation</td>
</tr>
<tr>
<td>l</td>
<td>Complement left of head</td>
</tr>
<tr>
<td>r</td>
<td>Complement right of head</td>
</tr>
<tr>
<td>t</td>
<td>Gapped topicalisation</td>
</tr>
<tr>
<td>T</td>
<td>Non-gapped topicalisation</td>
</tr>
</tbody>
</table>

Table 4.8: Markers added in the marking algorithm

VP is a valid phrasal category, WV is a valid word-level category, and WV is the first child of the VP. Therefore, the matching markers are applied to node and its children, resulting in:

(16)

\[
\text{VP} \quad \text{VV:h} \quad \text{给} \quad \text{NP:r} \quad \text{他们} \quad \text{3PL} \quad \text{QP:r} \quad \text{三块钱} \quad \text{3 dollars}
\]

4.6.3 BINARISE(T)

In accordance with the head structure made explicit by the marking algorithm, the trees are binarised so that heads seek their arguments, and are sought by adjuncts. In Chinese, this leads to left-branching structures in VPs (a head-initial phrase), and right-branching structures in NPs, ADJP s and ADVP s (all head-final phrases).

While this binarisation procedure is common to both Chinese and English CCGbank, an additional procedure is necessary for Chinese CCGbank, due to the structural distinction between phrase- and word-level tags which allows PCTB to encode head structure.

```
<table>
<thead>
<tr>
<th>NP</th>
<th>ADVP</th>
<th>ADJP</th>
<th>QP</th>
<th>CLP</th>
<th>DP</th>
<th>VP</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>RB</td>
<td>JJ</td>
<td>CD</td>
<td>M</td>
<td>DT</td>
<td>VV</td>
</tr>
</tbody>
</table>
```

Figure 4.9: Typical unary projections

Because phrase-level tags can never occur at leaf nodes, PCTB contains many unary projections from word-level to phrase-level tags, including those in Figure 4.9. The full list of
Algorithm 3 Marking algorithm, \texttt{MARK}(T)

Precondition: a preprocessed \texttt{PCTB} tree \(T\)

1. \textbf{for} child \(\in\) node \textbf{do}
2. \hspace{1em} \textbf{match} child \textbf{do}
3. \hspace{2em} \textbf{try} \texttt{XP-TPC}:t \hspace{1em} \triangleright \text{-TPC functional tag with an index is gapped topicalisation}
4. \hspace{2em} \textbf{try} \texttt{XP-TPC}:T \hspace{1em} \triangleright \text{-TPC functional tag without an index is non-gapped topicalisation}
5. \textbf{for} node \(\in\) \texttt{NODES}(T) \textbf{do}
6. \hspace{1em} \textbf{match} node \textbf{do}
7. \hspace{2em} \textbf{try} \texttt{PRN:p} \hspace{1em} \triangleright \text{parentheticals are headed by the opening punctuation token}
8. \hspace{3em} \texttt{*:h}
9. \hspace{2em} \textbf{try} \texttt{IP}
10. \hspace{3em} \texttt{XP-SBJ:l \texttt{VP:h}} \hspace{1em} \triangleright \text{predication is headed by the VP}
11. \hspace{2em} \textbf{try} \texttt{VSB then VCP then VRD} \hspace{1em} \triangleright \text{verb compounds}
12. \hspace{3em} \texttt{*:h \texttt{*:r \texttt{*:a \texttt{*:h \texttt{*:r}}}}}
13. \hspace{2em} \textbf{try} \texttt{XP}
14. \hspace{3em} \texttt{X:c CC|PU X:c \ldots \texttt{ETC:&}} \hspace{1em} \triangleright \text{mark each conjunct in coordination}
15. \hspace{2em} \textbf{try} \texttt{UCP}
16. \hspace{3em} \texttt{X:C CC|PU Y:C \ldots \texttt{ETC:&}} \hspace{1em} \triangleright \text{mark each conjunct in UCP}
17. \hspace{2em} \textbf{try} \texttt{HP}
18. \hspace{3em} \texttt{H:h \texttt{*:r}} \hspace{1em} \triangleright \text{head-initial complementation}
19. \hspace{2em} \textbf{try} \texttt{HP}
20. \hspace{3em} \texttt{*:l \texttt{H:h}} \hspace{1em} \triangleright \text{head-final complementation}
21. \hspace{2em} \textbf{try} \texttt{XP-APP:A \ldots \texttt{*:r}} \hspace{1em} \triangleright \text{apposition}
22. \hspace{2em} \textbf{if} node \text{ is an argument cluster} \textbf{then}
23. \hspace{3em} \texttt{*} \hspace{1em} \triangleright \text{mark each argument in an argument cluster}
24. \hspace{3em} \texttt{*:@}
25. \hspace{2em} \textbf{else}
26. \hspace{3em} \texttt{*} \hspace{1em} \triangleright \text{last resort: mark as head-final}
27. \hspace{3em} \texttt{*:l \texttt{*:h}}
Unary projections transformed by the conversion algorithm can be found in Appendix A. However, since CCG grammars do not distinguish between word- and phrase-level categories, these unary projections must be collapsed before we label the nodes of each tree with CCG categories. Collapsing a unary projection involves replacing the node labelled with a phrase-level category with its (single) child labelled with a word-level category; the parent also inherits its child’s marker. The collapsing operation must be performed after marking (since the phrase-/word-level tag distinction is used to induce head structure), but before labelling.

**Attaching final punctuation high**

In any node whose rightmost children are all punctuation tokens, BINARISE will attach the punctuation as high as possible:

\[
\begin{align*}
\text{(17)} & \quad \begin{array}{c}
\text{XP} \\
X & \text{AP} & \text{PU} & \text{PU} \\
\end{array} \quad \rightarrow \quad \begin{array}{c}
\text{XP} \\
X & \text{AP} & \text{PU} & \text{PU} \\
\end{array} \\
\end{align*}
\]

The effect of attaching punctuation as high as possible is to reduce the number of punctuation absorption rules in the corpus. As such, the punctuation absorption rules will mention top-level categories but not the profusion of categories which may occur at any other internal level of the derivation.

**Hoisting paired punctuation**

If any non-parenthetical node in the tree has the paired punctuation structure, where \( \langle \) and \( \rangle \) represent any pair of the paired punctuation tokens in Table 4.9, then BINARISE hoists the paired punctuation in the following way:

\[
\begin{align*}
\text{(18)} & \quad \begin{array}{c}
\langle \text{XP} \rangle \\
\end{array} \quad \rightarrow \quad \begin{array}{c}
\text{XP} \\
\text{XP} \\
\end{array} \\
\end{align*}
\]
4.6. The Chinese CCGbank conversion algorithm

<table>
<thead>
<tr>
<th>Pair</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>“”</td>
<td>Double quotes</td>
</tr>
<tr>
<td>「」</td>
<td>Square quotes</td>
</tr>
<tr>
<td>()</td>
<td>Full-width parentheses</td>
</tr>
<tr>
<td>( )</td>
<td>ASCII parentheses</td>
</tr>
<tr>
<td>» «</td>
<td>Single quotes</td>
</tr>
<tr>
<td>« »</td>
<td>Double angle brackets</td>
</tr>
<tr>
<td>「」</td>
<td>Double square quotes</td>
</tr>
<tr>
<td>〈 〉</td>
<td>Angle brackets</td>
</tr>
<tr>
<td>＜ ＞</td>
<td>Full-width less/greater than signs</td>
</tr>
</tbody>
</table>

Table 4.9: Paired punctuation

4.6.4 \textsc{catlab}(T)

At this stage in the conversion, \textsc{pctb} trees have been reshaped into binary-branching trees in accordance with head structure. We are now ready to assign \textsc{ccg} categories to the nodes in a top-down fashion. As in Hockenmaier and Steedman (2007), labelling begins by mapping the \textsc{pctb} tag of the root to a \textsc{ccg} category. The root mapping for Chinese CCGbank is given in Table 4.10.

Then, \textsc{catlab} is called on the root, matching the marker configuration of a node and its children against the schemata in Table 4.11, applies the matching category labelling to its children, and calls \textsc{catlab} recursively on the now-labelled children. Line 2 of the algorithm is responsible for injecting \textsc{ccg} atoms into the resulting labelled derivation, by mapping selected tags from the \textsc{pctb} tagset to \textsc{ccg} atoms. For example, to inject the atom \textsc{qp}, the labelling procedure maps \textsc{pctb} tags of the form $\textsc{qp}^*$ to the atom \textsc{qp}.

Line 9 of the algorithm performs the fallback labelling, which is left adjunction, when none of the schemata in Table 4.11 match. This condition occurs in 995 (3.56%) of \textsc{pctb} trees.

Following category labelling, nodes have been labelled with \textsc{ccg} categories. However, constructions involving movement phenomena such as topicalisation and extraction must be reshaped into the trace-free analyses required by \textsc{ccg}.
Algorithm 4 Category labelling algorithm, $\text{CATLAB}(node)$

1. if $node$ does not have a category then
   2. map node's POS tag to a category
3. if $node$ is a leaf then
   4. return $node$
5. match $node$ against the schemata in Table 4.11 in turn
6. if a matching schema is found then
   7. label $node$ using the matching schema
   8. else
   9. label $node$ as per left adjunction $\triangleright$ treat unrecognised configurations as left adjunction
10. call $\text{CATLAB}$ recursively on the children of $node$
11. return $node$

<table>
<thead>
<tr>
<th>PCGPOS tag</th>
<th>Meaning</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>Sentence</td>
<td>$S[dcl]$</td>
</tr>
<tr>
<td>CP</td>
<td>Sentence with complementiser</td>
<td>$S[dcl]$</td>
</tr>
<tr>
<td>CP-Q</td>
<td>Yes-no question</td>
<td>$S[q]$</td>
</tr>
</tbody>
</table>

Table 4.10: Root mapping for Chinese CCGbank
Chapter 4 describes the aspects of the PCTB annotation style which lend it to corpus conversion. One of these is its analysis of constructions which yield non-local dependencies through traces — phonologically unrealised elements which mark the original sites of constituents moved or deleted under various constructions. However, the CCG Principle of Adjacency, a constraint on the form on combinatory rules, requires that CCG rules apply only over phonologically realised entities (Steedman, 2000). As a result, we project instances of constructions analysed in the PCTB through traces and movement to trace-free CCG analyses which preserve the dependencies indicated by the original annotation.

The skeleton of the fix algorithm is given in Algorithm 5.

<table>
<thead>
<tr>
<th>Relation</th>
<th>Schema</th>
<th>Relation</th>
<th>Schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predication</td>
<td>C</td>
<td>Left absorption</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>L \ C\L</td>
<td></td>
<td>PU \ C</td>
</tr>
<tr>
<td>Left adjunction</td>
<td>C</td>
<td>Right absorption</td>
<td>C \ PU</td>
</tr>
</tbody>
</table>
|                   | C/\:
| Right adjunction  | C, C:       | Coordination     | C: C[conj]   |
|                   | a            |                  | C[conj]      |
| Head-initial      | C            | Partial coordination | conj, C:c   |
|                   | C/\:
| Head-final        | L \ C\L:h   | Apposition       | XP, A, NP    |

Table 4.11: Category labelling schemata

4.6.5 \textbf{\texttt{fix}(T)}

Algorithm 5 Trace elimination algorithm, \texttt{fix}(T)

\textbf{Precondition:} labelled tree $T$, map from \texttt{TGREP} patterns to repair procedures \texttt{patterns}

\begin{algorithm}
\begin{algorithmic}
\State \textbf{for} $node \in \text{nodes}(T)$ \textbf{do}
\State \textbf{for} $pattern, fixer \in \text{patterns}$ \textbf{do}
\State \If {pattern matches node} \textbf{then}
\State apply fixer to node
\EndIf
\EndFor
\EndFor
\end{algorithmic}
\end{algorithm}
In our implementation of the corpus conversion pipeline, we use a variant of TGREP2 (Rohde, 2004) to match against local tree structures, optionally binding particular nodes to names. These names can be used to perform surgery on the labelled trees in order to reshape them into the analyses described in Chapters 2 and 3. This tree surgery technique is introduced in more detail in Section 5.1.

**Interactions between constructions**

CCG is modular, in the sense that the viability of a construction is determined only by the categories involved. For example, the Chinese relative clause construction is “powered” by the category \((NP/\ast)/(S[\text{dcl}]/NP)\) of the relativiser 的 de. It is the category of 的 de which places syntactic requirements on the categories of the other constituents which participate in the construction — namely, a gapped sentence \(S[\text{dcl}]/NP\) and an \(NP\). Because of this modularity, any span from which the category \(S[\text{dcl}]/NP\) can be derived is a candidate argument to the relative clause construction.

This category \(S[\text{dcl}]/NP\) can be derived in more than one way. A simple active declarative sentence with a subject gap (e.g. 喝啤酒 drinks beer) is one:

\[(19) \quad \text{a person who drinks beer} \]

\[
\begin{array}{cccc}
\text{喝} & \text{啤酒} & \text{的} & \text{人} \\
\text{drink} & \text{beer} & \text{DE} & \text{person} \\
\hline
(S[\text{dcl}]/NP)/NP & NP & (NP/NP)/(S[\text{dcl}]/NP) & NP \\
\hline
S[\text{dcl}]/NP & NP/NP & NP \\
\hline
NP/NP & NP
\end{array}
\]

Regardless of the internal structure of the gapped sentence complement, as long as it yields the category \(S[\text{dcl}]/NP\), the grammar predicts that the result is grammatical.

\[(20) \quad \text{the criminal who got arrested} \]

\[
\begin{array}{cccc}
\text{被} & \text{逮捕} & \text{的} & \text{犯人} \\
\text{BEI} & \text{arrest} & \text{DE} & \text{criminal} \\
\hline
(S[\text{dcl}]/NP)/(S[\text{dcl}]/NP)/NP & (S[\text{dcl}]/NP)/NP & (NP/NP)/(S[\text{dcl}]/NP) & NP \\
\hline
S[\text{dcl}]/NP & NP/NP & NP \\
\hline
NP/NP & NP
\end{array}
\]
Because sentences can contain multiple NLD types, the order in which the trace elimination procedures are performed affects the resulting analysis. Table 4.12 gives the trace elimination procedures used in the conversion process, and where multiple procedures can apply at the same time, the order in which they are performed. The following ordering rule determines the order in which the procedures should be applied:

**Observation 1.** If construction X can produce a result used in construction Y, the procedure for the analysis of construction X should precede the procedure for construction Y.

For instance, the 被 bei construction can yield the result $S[dcl]\backslash NP$ which can serve as a complement to the relative clause construction. Therefore the procedure which applies the CCG analysis of 被 bei should precede the one which applies the analysis of the relative clause construction.

In the same way, subject pro-drop on a transitive verb can yield the result $S[dcl]/NP$, which can serve as the object-gapped clause in the relative clause construction, so the re-analysis of pro-drop should precede the analysis of the relative clause construction.

**Category repair**

A key procedure in the trace elimination algorithm adjusts categories after tree reshaping operations, to ensure that the resulting trees remain valid normal-form CCG derivations. For instance, in Figure 4.10a, the trace elimination algorithm creates the dotted edge, leading to the tree in Figure 4.10b.

However, this results in the non-CCG rule $NP \ (S[dcl]\backslash NP)/NP \rightarrow S[dcl]$. Category repair applied to the subtree type-raises (> T) the NP and applies composition (> B) to produce the expected parent category $S[dcl]/NP$, restoring the derivation’s validity. Similar procedures apply other CCG combinators to repair the categories of intermediate trees.

![Figure 4.10: Intermediate trees may require category repair](image-url)
<table>
<thead>
<tr>
<th>Procedure</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long 被 bei</td>
<td>(21) 我的钱包被他抢走了 1SG DE wallet B EI 3 SG snatch-away ASP I had my wallet stolen.</td>
</tr>
<tr>
<td>Short 被 bei</td>
<td>(22) 城市被雨困住 city B EI rain imprison-firm The city was besieged by rain.</td>
</tr>
<tr>
<td>Gapped 把 ba</td>
<td>(23) 我把那封信送给 你 1SG BA that MW letter send-to 2 SG I sent that letter to you.</td>
</tr>
<tr>
<td>Gapped topicalisation</td>
<td>(24) 那部电影我还没看 1SG still have-not see That film, I still haven’t seen.</td>
</tr>
<tr>
<td>Non-gapped topicalisation</td>
<td>(25) 水果我喜欢哈密瓜 fruit 1 SG like type.of.melon As for fruit, I like yellow rockmelons.</td>
</tr>
<tr>
<td>Argument cluster coordination</td>
<td>(26) 我给你三块钱，你 1SG give 2 SG 3 MW money，2 SG little.brother 2 MW money I’ll give you three dollars, and your brother two dollars.</td>
</tr>
<tr>
<td>Subject extraction</td>
<td>(27) 卖蔬菜的人 sell vegetables DE person a person selling vegetables</td>
</tr>
<tr>
<td>Object extraction</td>
<td>(28) 人家卖的蔬菜 others sell DE vegetables vegetables that someone sells</td>
</tr>
<tr>
<td>pro-drop</td>
<td>(29) pro卖的蔬菜 pro sell DE vegetables vegetables that pro sells</td>
</tr>
</tbody>
</table>

Table 4.12: Ordering of trace elimination procedures
Algorithm 6 Category repair algorithm \texttt{fixcat(node, until)}

Precondition: starting node \texttt{node}, termination node \texttt{until}

\begin{algorithm}
\begin{algorithmic}
\While {node \neq until}
\If {node is not a binary rule}
\State continue
\EndIf
\State L, R, P \leftarrow \text{the context of node}
\If {L \rightarrow P is not a recognised rule}
\If {L \rightarrow Y \rightarrow X \rightarrow Y}
\State \Comment{apply \(> T\) to L}
\ElseIf {R \rightarrow Y \rightarrow X \rightarrow Y}
\State \Comment{apply \(< T\) to R}
\EndIf
\ElseIf {conj \rightarrow P}
\State conj \rightarrow R[conj]
\ElseIf {L \rightarrow R[conj] \rightarrow P}
\State L \rightarrow L[conj] \rightarrow L
\ElseIf {L is an atomic category}
\If {R \rightarrow P}
\State R \rightarrow R[conj]
\ElseIf {R \rightarrow R}
\State R \rightarrow R
\ElseIf {R is an atomic category}
\If {L \rightarrow P}
\State L \rightarrow L
\Else
\If {T/(T/X) \rightarrow (T/X)/(T\backslash A)}
\State \Comment{apply \(> T\) to A}
\ElseIf {X \rightarrow Y \rightarrow X \rightarrow Y}
\State \Comment{apply \(< T\) to A}
\EndIf
\ElseIf {S[dcl]\backslash X} \rightarrow S[dcl]\backslash X}
\State \Comment{generalise modifier}
\Else
\State \Comment{try all composition rules}
\EndIf
\EndIf
\EndIf
\EndIf
\EndIf
\EndIf
\EndAlgorithm
\end{algorithm}

\texttt{node} \leftarrow \texttt{node.parent}
The category repair algorithm in Chinese CCGbank (Algorithm 6) is similar to the one in Honnibal (2010) for propagating hat-ccg labelled categories through a derivation. Honnibal (2010) discusses two ways for propagating changes through a derivation: top-down or bottom-up, depending on the direction in which category changes are pushed through the derivation. While the category repair algorithm in Honnibal (2010) operates top-down, ours operates bottom-up, because changes to categories required by the Chinese CCGbank re-analyses are generally at the leaves of the derivation.

Each if-statement (as in Line 6, 8, 10 and so on) of the category repair algorithm tries to unify the categories of the node and its children with the given pattern (e.g. $L (X|L)|Y \rightarrow X|Y$). If unification succeeds, the algorithm rewrites the categories of that node and its children according to the consequent (Line 7).

For instance, the local context in (30-a), which induces the unrecognised ccg rule $NP (S[decl]\backslash NP)/NP \rightarrow S[decl]/NP$, is unifiable with the context in the branch in Line 6. To fix the derivation, the algorithm applies type-raising (Line 7), yielding the valid subtree in (30-b).

$$
(30) \quad \begin{array}{c}
\text{a.} \\
S[decl]/NP \\
NP \quad (S[decl]\backslash NP)/NP \\
\text{b.} \\
S[decl]/NP \\
S/(S\backslash NP) \quad (S[decl]\backslash NP)/NP \\
NP
\end{array}
$$

The algorithm proceeds in this way, bottom-up, until the designated node until, marking the top node of a construction, is reached.

**Generalising modifier categories**

In Section 2.5.1, we described the use of backward crossed composition ($<B_c>$) to generalise over the categories of aspect particles, allowing a single category to modify verbs with any number of rightward arguments. To enact this analysis, we apply Algorithm 7.
Algorithm 7 Generalising over modifier categories

Precondition: Labelled root root

1. for node ∈ NODES(root) do
2.     L, R, P ← the context of node
3.     if \( L R \rightarrow P \) is a candidate for \( \langle B'_n \rangle \) then
4.         node.cat ← BXCOMP-GEN\((n, L, R)\)
5.     else if \( L R \rightarrow P \) is a candidate for \( \langle B_n \rangle \) then
6.         node.cat ← BXCOMP\((L, R)\)

4.7 Summary

This chapter has described the annotation characteristics of Penn Chinese Treebank and CCGbank, the two corpora most relevant to this work. We have described how the distinction between phrase- and word-level tags allows the PCTB to encode head structure and the complement/adjunct distinction. This allows Chinese CCGbank to exploit gold-standard annotation, eschewing the use of head-finding heuristics used by Hockenmaier (2003) for English CCGbank.

We have presented the framework of the Chinese CCGbank conversion algorithm, which demonstrates that these gold-standard head annotations can be adapted for use in the CCGbank conversion pipeline developed by Hockenmaier (2003). The CCGbank pipeline consists of a marking step, annotating constituents with head structure, a binarisation on the \( k \)-way branching trees, category labelling which recursively assigns \( \text{ccg} \) categories to nodes according to head structure, followed by re-analysis, in which constructions annotated with traces and movement are transformed into trace-free \( \text{ccg} \) analyses.

This final stage is the focus of the next chapter, and a substantial challenge in the production of a corpus conversion. We are now ready to describe how Chinese CCGbank reshapes the annotations of NLD-generating syntax in the PCTB, in order to produce the trace-free \( \text{ccg} \) analyses of Chapter 3.
Chapter 5

Extracting NLD analyses for Chinese CCGbank

前人種樹，後人乘涼。

One generation plants the trees; the next enjoys the shade.

The previous chapter introduced the key resources for our corpus conversion, and presented the framework of the Chinese CCGbank conversion algorithm which incrementally creates CCG derivations from Penn Chinese Treebank trees. Just as Chapter 3 focused on the abstract analysis of NLD-generating syntax, this chapter is dedicated to describing the procedures that re-shape the PCGB analysis of NLD syntax through traces and movement into equivalent trace-free CCG derivations which yield the same dependencies.

We adapt the treebank pattern matching language TGREP2 (Rohde, 2004) to find instances of NLD-generating syntax, and perform local tree transformations to achieve the re-analyses presented in Chapter 3. Finally, we examine the sparsity characteristics of the resulting corpus, which are a measure of how well the analysis of syntax in the corpus generalises properly over syntactic phenomena.

This work was presented in 2010 at the 23rd International Conference on Computational Linguistics (COLING) in Beijing, China (Tse and Curran, 2010).
5.1 Traces

In the instantiation of $X'$ syntax in the Penn Chinese Treebank, whenever a syntactic phenomenon involves deletion or movement, the canonical position of the deleted or moved constituents remain *in situ* in the derivation as phonologically unrealised elements, known as *traces* or *empty categories* (ECs). In the annotation, all empty categories have the POS tag `-NONE-`, while the type of trace (and coindexation if any) appears where a lexical item would occur in an overt constituent. The trace types which occur in the *PC* annotation are given in Figure 5.1. The distinction between $A$-movement and $A'$-movement is internal to the instantiation of government and binding theory used by the annotation.¹

<table>
<thead>
<tr>
<th>Trace type</th>
<th>lexical item</th>
<th>description</th>
<th>occurs in</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRO</td>
<td><em>PRO</em></td>
<td>Null pronominal in non-finite clause</td>
<td>control/raising</td>
</tr>
<tr>
<td>pro</td>
<td><em>pro</em></td>
<td>Null pronominal</td>
<td>pro-drop</td>
</tr>
<tr>
<td>$A'$ trace</td>
<td><em>T</em></td>
<td>Trace of $A'$ movement</td>
<td>extraction</td>
</tr>
<tr>
<td>$A$ trace</td>
<td>*</td>
<td>Trace of $A$ movement</td>
<td>passivisation</td>
</tr>
<tr>
<td>RNR trace</td>
<td><em>RNR</em></td>
<td>Trace of right node raised constituent</td>
<td>RNR</td>
</tr>
</tbody>
</table>

Table 5.1: Trace types in the Penn Chinese Treebank annotation

Because Steedman (2000)'s Principle of Adjacency (Principle 1) states that combinatorial rules may only apply to phonologically realised, string-adjacent entities, a great proportion of the special constructions handled by our conversion algorithm revolves around transforming instances of constructions whose PC annotations involve traces into trace-free CCG analyses which are equivalent in a way we will describe.

This phase of the Chinese CCGBank conversion pipeline is related to the re-entrancy annotation phase of the LFG corpus conversion procedure of Cahill et al. (2004) and adapted by Guo et al. (2007a) for Chinese, which maps traces in *c*-structure to re-entrancies in *f*-structure (Figure 5.1). This procedure in Cahill et al.'s algorithm replaces gapped arguments with re-entrancies (□) which are co-indexed with a local $AVM$ sub-structure (for instance the grammatical function *topic* in Figure 5.1b).

Guo et al. (2007b) also observed that unlike English, nearly 70% of antecedents in Chinese NLDs are not co-indexed with any antecedent (including the non-gapped constructions we cover in §§ 3.3, 3.4, 3.2.1). To account for this, they adapt Cahill et al.'s algorithm.

¹ $A$-movement is movement to an $A$-position (an argument position), and $A'$-movement is movement to an $A'$-position (a non-argument position, like Spec).
by modifying the ranking function used to select the correct target for a re-entrancy. Instead of conditioning on the grammatical function of the (possibly absent in Chinese) antecedent as in Cahill et al. (2004), the ranking function conditions on the grammatical function of the trace.

Just as the LFG procedures connect a trace with its target through a path in the \( f \)-structure, our tree reshaping procedures rewrite categories on a path between the trace and its target to reflect the presence of a trace.

![Diagram](image)

(a) \( c \)-structure with trace

(b) \( f \)-structure with re-entrancy

Figure 5.1: In LFG, traces in \( c \)-structure correspond to re-entrancies in \( f \)-structure

We describe each of the trace types occurring in the source corpus, their distribution in Chinese, and describe the transformation procedure which enacts the Chinese CCGbank analysis.

### 5.1.1 Describing transformation procedures

To describe the tree transformations which project \( PCTB \)-structured analyses onto trace-free \( CCG \) analyses, we needed a way to locate instances of each kind of NLD syntax, and address particular nodes in the annotation so that their structure could be modified.

\textsc{tgrep2} (Rohde, 2004) is a query language over \( PTB \)-style constituent trees. Each expression of the query language is a filter over the structure of a tree, allowing users to select trees or sub-trees of interest in a corpus. \textsc{tgrep2} has been used to extract features to predict L2 readability (Heilman et al., 2007), retrieve collocations from parsed trees (Fazly and Stevenson, 2006) and find corpus examples for manual analysis (Weber and Müller, 2004).

\textsc{tgrep2} expressions specify constraints on the structure of matching trees. For instance, the expression \( \text{NP} \leftarrow \text{PP} \). \text{VV} matches nodes \( \text{NP} \) which dominate a \text{PP} followed by a \text{VV}. The
expressions can grow arbitrarily complex, allowing them to pattern-match against the structure of arbitrary sub-trees.

Tgrep2 allows queries over a dataset, but not updates. We adapted the approach of Levy and Andrew (2006), who developed tsurgeon: a dialect of Tgrep2 which is capable of “tree surgery” operations of the kind that we perform to restructure pctb trees into CCG derivations. They augment Tgrep2 with the ability to bind names to particular matched nodes. For instance, the expression in (1-a) binds the name np to the boxed node in (1-b). A small set of tree surgery operations can then address the bound nodes to effect changes in tree structure.

(1)  
\[ \text{a. } PP < (NP=np < (ADJP \text{ $+ NP$} \text{ $+ PP$})) \]
\[ \text{b. } \]
\[ \begin{array}{c}
\text{PP} \\
\text{NP} \\
\text{ADJP} \\
\text{NP} \\
\text{PP}
\end{array} \]

Given the complexity of some of the tree reshaping procedures required by the Chinese CCGbank analysis, we needed to augment this approach to allow the execution of arbitrary code to modify a local tree structure, rather than restrict ourselves to a small set of tree surgery primitives as Levy and Andrew (2006) do. Furthermore, we wanted a tsurgeon analogue which could handle CCGbank and pctb data natively.

To this end, we developed our own tree manipulation library, munge, which adapts the syntax of Tgrep2. Munge shares the ability of tsurgeon to bind names to particular tree nodes. However, munge then allows arbitrary Python code to manipulate these nodes, effecting arbitrary tree transformations.

Because the munge expressions can grow quite complex, the query expressions used to select candidates for tree surgery in this chapter are presented in graphical form. For the corresponding expressions, the user can consult the Chinese CCGbank codebase.

The repair procedures assume that nodes have a parent pointer parent and a category field cat; complex categories have two fields res and arg for their result and argument category respectively.

Finally, the repair procedures often invoke the shrink operation, which excises a trace:

---

The sub-expression NP < (ADJP $+ NP) matches any subtree with label NP with a descendant with label ADJP (the operator < selects descendants), with a sibling immediately to the right with label NP (the operator $+ selects immediate right siblings). For a full list of the tsurgeon operators, consult Levy and Andrew (2006).
### 5.1. Traces

<table>
<thead>
<tr>
<th>TGREP expression</th>
<th>Graphical representation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>VP &lt; { VV $ NP-OBJ }</td>
<td><img src="image" alt="VP diagram" /></td>
<td>Select VP nodes which immediately dominate a VV which has an NP-OBJ sibling.</td>
</tr>
<tr>
<td>VP &lt;&lt; ^/*T*-d+/</td>
<td><img src="image" alt="VP diagram" /></td>
<td>Select VP nodes which dominate a node with lexical item <em>T</em>-i for some index i.</td>
</tr>
<tr>
<td>IP &lt; NP-SBJ=N &lt; VP=V</td>
<td><img src="image" alt="IP diagram" /></td>
<td>Select any node with NP-SBJ and VP children, binding the names N and V to the respective matched children, and binding the name TOP to the matched node.</td>
</tr>
<tr>
<td>*/-SBJ$/a</td>
<td>*-SBJ</td>
<td>Match any node whose tag ends in the functional tag -SBJ.</td>
</tr>
</tbody>
</table>

| Table 5.2: Graphical presentation of MUNGE expressions |

5.1.2 **pro traces**

The trace type *pro* marks the canonical position of an argument deleted under pro-drop. To instantiate the unary rule analysis of pro-drop (Rule 12), we remove the node dominating the trace node, leaving the node PP a unary projection as desired.
5.1.3 PRO traces

The trace type *PRO* indicates the obligatorily deleted subject of an embedded non-finite clause, under identity with another argument position. PRO is implicated in control and raising structures such as:

(4) a. I encouraged him to PRO enrol in geology.
   b. China encourage private entrepreneur invest national basic construction
     China encourages private entrepreneurs, to PRO invest in national infra-structure

In classical gb theory, PRO receives the feature matrix $\begin{bmatrix} +\text{anaphoric} \\ +\text{pronominal} \end{bmatrix}$, lending it a split personality. In (5-a), PRO is controlled by the main verb encourage, such that its reference depends on the main verb subject Poirot (“controlled PRO” in Chomsky (1993)). In (5-b), the interpretation of PRO is not contingent on any constituent in the sentence (“arbitrary PRO” in Chomsky (1993))). These two respective states of affairs are the anaphoric (5-a) and pronominal senses (5-b) of PRO.

(5) a. Poirot was glad [ [ PRO to abandon the investigation. ] IP ] CP
   b. [ [ PRO To abandon the investigation ] IP ] CP would be a mistake.
     (from Haegemann (1994))

English has two clausal types which are headed by non-finite verbs: infinitivals (e.g. to abandon the investigation), and gerunds (e.g. abandoning the investigation) (Xu, 1986). The former exhibit obligatory PRO, where it is ungrammatical for the non-finite VP’s subject to exhibit an overt subject:

(6) a. Poirot was glad [ [ PRO to abandon the investigation. ] IP ] CP
   b. *Poirot was glad [ [ Poirot to abandon the investigation. ] IP ] CP

On the other hand, the subject of a gerund may be overt, or it may be a PRO with arbitrary reference:

(7) a. [ [ PRO Abandoning the investigation ] IP ] CP was the only thing left to do.
   b. [ [ Poirot abandoning the investigation ] IP ] CP was the only thing left to do.
5.1. Traces

<table>
<thead>
<tr>
<th>Argument type</th>
<th>Annotation</th>
<th>Referent</th>
<th>Replaceable with overt subject?</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overt subject</td>
<td>XP-SBJ</td>
<td>-</td>
<td>-</td>
<td>event</td>
</tr>
<tr>
<td>pro trace</td>
<td><em>pro</em></td>
<td>discourse</td>
<td>Yes</td>
<td>event</td>
</tr>
<tr>
<td>PRO trace</td>
<td><em>PRO</em></td>
<td>arbitrary</td>
<td>No</td>
<td>predicate</td>
</tr>
</tbody>
</table>

Table 5.3: Types of subjects of non-finite IPs in PCTB

Arbitrary PRO

The gerund form of an English verb is accompanied by a change in the verb's form:

(8)  a. Baking a rhubarb and custard pie is not as easy as it appears.
     b. *Bake a rhubarb and custard pie is not as easy as it appears.

In Chinese, any non-finite IP may function as a NP without the need for marking. Thus, the subject of 保持 maintain is the NP 中国 economy in (9-a), and the IP 丹东利用 外资 Dandong use foreign investment in (9-b).

(9)  a. 中国 经济 保持 高速 增长
     Chinese economy maintain fast increase
     Chinese economy maintains rapid growth
    b. 丹东 利用 外资 保持 高速 增长
     Dandong use foreign investment maintain fast increase
     Dandong’s use of foreign investment maintains rapid growth

In Chinese, there are three alternatives for the realisation of the subject slot in this construction, summarised in Table 5.3. We now consider the two possibilities in which the subject slot is filled by an empty category (pro or PRO).

(10)  a. *pro* 救援 大熊猫 这 一 珍稀 濒危 物种 ，依然
      pro rescue panda this one rare endangered species， still
      heavy burden．
     Rescuing the rare, endangered panda is critical and there is still a long way to go.
    b. *PRO* 学 英文 是 世界 潮流
      PRO learn English is world trend
      Learning English is a global trend
The distinction between (10-a) and (10-b) is twofold. First, pro in (10-a) is referentially dependent on a discourse subject (here, the rescuers), while PRO in (10-b) has arbitrary reference. Secondly, the IP in (10-a) describes an event — the discourse subject of rescuing pandas — while the IP in (10-b) describes a predicate — the act of learning English. Because 学英文 learning English describes a predicate, its subject slot cannot be occupied by an overt agent argument.

In Chinese CCGbank, both pro and PRO subjects of non-finite IPs are treated uniformly:

**Rule 13** (Subject-dropping rule).

\[(S[\text{dcl}\backslash NP])_1 \rightarrow S[\text{dcl}]_1\]

Because a non-finite IP with a non-overt subject is ambiguous between pro and arbitrary PRO, we allow all IPs to undergo subject-dropping.

**Controlled PRO**

The control theory of GB proposes two distinctive properties for controlled PRO:

1) PRO is not substitutable with an overt subject

2) the PRO subject of the embedded clause is obligatorily identified with (i.e. controlled by) some argument of the main verb in a lexically specified manner.

Controlled PRO cannot be replaced by an overt subject, demonstrating property 1:

(11)  a. 中国 设法 *PRO* 发展 科技  
    China try PRO develop technology  
    China attempts to develop technology

    b. *中国 设法 中国 发展 科技  
    China try China develop technology

The subject of the embedded verb is co-referent with some argument of the main verb, demonstrating property 2:

(12)  a. 张三 权 *PRO* 参加 会见  
    Z. encourage L. PRO join meeting  
    Zhangsan encourages Lisi to participate in the meeting
b. 张三答应我*PRO*会参加会见
Zhangsan replied to me that he will participate in the meeting

In our ccg analysis for controlled PRO, the embedded IP corresponds to an \( S[dcl]\ NP \) argument which is coindexed with other arguments of the main verb category according to the type of control it selects for. The correspondence between a VP with controlled PRO and the category \( S[dcl]\ NP \) can be seen in Figure 5.2.

### 5.1.4 Traces of wh-movement

Wh-movement is implicated in several Chinese syntactic constructions including relativisation and topicalisation.\(^3\) Each of these constructions involves an A-position being moved to a A’-position such as Spec.

#### Extraction

The PCTB account of extraction is a variation of the head external analysis of the relative clause construction developed by Chomsky (1977). The distinctive features of this analysis are that:

- the relative clause is a CP (complementiser phrase) adjoined to the head noun phrase
- the relative clause CP is headed by the relativiser (that in English, or ｂｙ de in Chinese)
- the extracted argument inside the CP undergoes movement to Spec of CP

In the PCTB annotation, the movement to Spec is indicated by coindexing the trace of the moved NP with the WHNP which immediately dominates its landing site. This coindexation allows for disambiguation of which trace underwent movement, if there are multiple traces inside the CP. WHNP dominates a node *OP*, a trace of the relative pronoun, which the PCTB annotation treats as unrealised. Finally, the coindexation between the head noun and the gap inside the CP is left implicit.

The normal PCTB annotation for subject extraction involves a WHNP operator co-indexed with a gap in the adnominal clause:

\(^3\) As its name implies, wh-movement was first studied in English, where according to the analysis of Chomsky (1977), wh-words move from their canonical positions to Spec of CP. While Chinese exhibits wh-in-situ, in which wh-words in wh-questions remain in their canonical position, an influential analysis first articulated by Huang (1982b) suggests that Chinese exhibits wh-movement at LF.
Figure 5.2: IP with controlled PRO subject corresponds to $S[\text{dcl}]\backslash NP$
(13) a thoroughfare which will link the north and south of the Yangtze

However, extraction is not consistently annotated in PC-TB, and some annotated instances of extraction lack the WHNP operator, and therefore lack the co-indexation with a gap in the adnominal clause. We refer to this configuration as a compact relative clause:

(14) a beneficial stage which is flourishing and stable
The algorithm proceeds the same way despite the inconsistency; the \textit{WHNP} trace is only shrunk when analysing full relative clauses. Algorithm 8 achieves the re-analysis of subject extraction.

\textbf{Algorithm 8 Re-analysis of subject extraction}

\textbf{Precondition:}

\begin{align*}
N \\
CP|\text{NP}|NP-PRD=PRED \\
WHNP-i=W \\
CP|\text{IP} \\
NP-SBJ \\
*T*=T
\end{align*}

1. \textbf{if} this is a compact relative clause \textbf{then}
2. \hspace{1em} shrink \textit{W}
3. \hspace{1em} extract trace index \textit{i} from \textit{W}
4. \hspace{1em} \textbf{for} every match of \textit{PP} under \textit{N} \textbf{do}
5. \hspace{2em} shrink \textit{T}
6. \hspace{2em} category repair from \textit{S} to \textit{N}
7. \hspace{2em} relabel relativiser category

The \textit{PCTB} annotation for object extraction is similar, differing in the location of the trace in the adnominal clause. In addition to direct objects (\textit{NP-OBJ}), extent complements of verbs such as \textit{实现} realise (analysed by the \textit{PCTB} as ditransitive verbs and annotated \textit{NP-EXT}) are also valid targets for extraction:
Algorithm 9 Re-analysis of object extraction

Precondition:

1. if this is a compact relative clause then
2. shrink \( W \)
3. extract trace index \( i \) from \( W \)
4. for every match of
5. shrink \( T \)
6. category repair from \( S \) to \( \text{TOP} \)
7. relabel relativiser category

(15) the number of slaves which people from rich households keep
The algorithm handles extraction of extent complements by licensing NP-EXT as well as NP-OBJ as potential gaps. The full algorithm for the re-analysis of object extraction is given in Algorithm 9.

In the discussion which follows, the Chinese relative clause construction is decomposed into three parts: the complement, the subordinator, and the head. In the following sentence, 张三刚买 Zhangsan bought 他 the complement, 的 de is the subordinator, and 蔬菜 vegetables 是 the head:

(16) 张三刚买蔬菜
Z. just buy the vegetables Zhangsan just bought

We distinguish the three types of extraction in Table 5.4. In subject and object extraction, the head is co-referent with the gap in the complement — subject if the gap is in subject position, and object if the gap is in object position.

Non-gapped extraction, on the other hand, does not involve the identification of the head with a gap in the complement. In the Penn Chinese Treebank annotation, non-gapped extraction is identified with non-gapped topicalisation (discussed earlier in Section 3.3), so that the second sentence is derived by movement from the first:

(17) a. 体育场 中国 援建 多哥
stadium China help-build Togo
As for the stadium, China helped Togo build it.

b. 中国 援建 多哥的体育场
China help-build Togo 蔬菜 stadium
the stadium that China helped Togo build

In CCG, the long-distance dependency between the head and gap in the relative clause construction is mediated by the category of the subordinator word. In non-gapped extraction, however, the trace in the PCtb annotation occurs in a non-argument position (namely Spec of IP), ruling out a direct dependency between, for example, 体育场 stadium and 援建 help to build in (17-a). In contrast, the long-distance dependency between 蔬菜 vegetables and 买 buy in (16) can be recovered because the object of 买 buy is an argument of the verb.

Gapped topicalisation

In the PCtb annotation for topicalisation, the topic and comment are siblings under IP. In the gapped case, the topic carries an index (e.g. NP-TPC-1), shared with the trace in the
Non-gapped extraction

In the PCTB annotation of non-gapped topicalisation, a NP-TPC node is adjoined to an IP or VP (depending on whether it represents displacement-to-S or displacement-to-VP).

(21) 

```
NP-TPC
  
NP-SBJ
  
IP
  
VP

这次比赛
this match

我国参赛的新选手
new players playing for China

many
```

We reshape instances of non-gapped topicalisation by inserting the unary rule \( NP \rightarrow S/S \) or \( NP \rightarrow VP/VP \) (depending on the level at which the topicalised phrase is adjoined) above the NP-TPC node, resulting in:
Figure 5.3: Chinese CCGbank analysis of gapped topicalisation
Algorithm 10 Re-analysis of gapped topicalisation

Precondition:

\[ \text{IP} \mid \text{CP-CND} = P \]
\[ \ast \text{-TPC}_i = T \quad \text{IP} \mid \text{CP-CND} = S \]

1. extract trace index \( i \) from \( T \)
   \[ S/(S/X) \]

2. replace \( X = T \) with \( X \)
   \( \triangleright \) Rule 9 (gapped topicalisation)

3. for every match of \( \text{IP} = \text{TOP} \) under \( S \) do
   \( \text{PP} \)
   \( \text{P} \)
   \( \text{NP-SBJ} \mid \text{NP-OBJ} = T \quad S \)
   \( \quad \ast \text{T}_* - i \)

4. shrink \( T \)

5. category repair from \( S \) to \( \text{TOP} \)

(22) As for this match, there are a lot of new team members playing for China.

这次比赛 我国参赛的新选手 许多
this match new players playing for China many

\[ \text{NP} \stackrel{T_{\text{gap}}}{\longrightarrow} \text{NP} \stackrel{S[dcl] \backslash \text{NP}}{\longrightarrow} \]

\[ \frac{S/S}{S[dcl]} \]

5.1.5 Right node raising

The term right node raising (RNR) refers to the sharing of one or more rightward arguments among one or more coordinated constituents (Clapp, 2008). For instance, in (23) the single argument 俄罗斯客人 Russian guests is shared by the two functors 会见 meet with and 宴请 invite to dinner.

(23) 会见 俄罗斯客人 与 邀请 俄罗斯客人
meet-see:res RNR and banquet-invite asp Russia guest
meet with, and invite to dinner, the Russian guests

(24) I could have planned, and Petra could have hosted, a huge party.
(from Clapp (2008))
Two RNR analyses from the literature, both described in Clapp (2008), are the *account by ellipsis* (25-a) and the *ATB* (across-the-board) *movement account* by Ross (1967) (25-b):

\[\text{(25) a.}\]

```
<table>
<thead>
<tr>
<th>XP</th>
</tr>
</thead>
<tbody>
<tr>
<td>_XP _conj _XP</td>
</tr>
<tr>
<td>X _t_1 \ldots _t_n</td>
</tr>
<tr>
<td>X _A_1 \ldots _A_n</td>
</tr>
</tbody>
</table>
```

In the account by ellipsis, the rightward arguments stay *in situ* in the last conjunct, and leave traces in the corresponding argument positions of each other conjunct. In the ATB movement account, the right node raised arguments are reattached at a position which commands all the conjuncts.

The analysis of right node raising in the PCTB resembles the account by ellipsis, where, in all but the last conjunct, each right node raised argument is replaced by a trace *RNR*-i which is coindexed with the corresponding argument in the last conjunct. In addition, coordination between bare verbs in the PCTB is also represented as RNR, in order to maintain its strict distinction between phrasal structure and compound-internal structure.

In contrast, the analysis of right node raising in CCG is more similar to the ATB movement account. Because CCG possesses a freer notion of constituency — any span of lexical items which can be formed into a unit by the combinatory rules are considered a constituent — the residues of the coordination of so-called non-constituents are well-typed, and can be coordinated using the regular coordination scheme (Rule 4) provided each conjunct shares the same category. Once the residues have been coordinated, they collect their arguments in the canonical manner.
We transform the account by ellipsis into a form amenable to re-analysis through CCG. The re-analysis begins by finding the node $\alpha$ which dominates all the rnr conjuncts. In Chinese, as in English, rnr of more than one rightward argument is possible:

(27)  use and reinvigorate community assets to the value of more than 10 billion yuan

We collect the indices of all the rnr traces, then shrink each trace. Then, we locate the actual argument corresponding to each rnr trace, and re-attach each in turn above the node $\alpha$, resulting in an analysis with the following structure:
Right node raising without a coordination word

Some instances annotated as right node raising in the PCTB lack the explicit coordination word which we use to identify cases of right node raising. We identified two distinct constructions both annotated as right node raising without a coordination word: one similar to verb coordination (VCD), and one in phrases of estimation. We argue that these are best analysed as structures different to right node raising, and that to analyse them correctly in Chinese CCGbank would require a reinterpretation of the original annotation. We first consider the VCD-like construction.

(29) demand and receive the money resulting from their work and securities

Because no coordination word occurs, and because the dependencies resulting from both constructions are the same, we suggest that these cases be unified with VCD (verb coordination), which should result in the morphologically pre-composed analysis in Section 2.9.4. However, we do not perform such a re-analysis in the final corpus.

Phrases of estimation

There are a small number of QPs which contain two numerals without an intervening coordination word, with the syntax \(N_1 + N_2 + MW\). This configuration expresses uncertainty about the precise value of the designated object:

(30) two three\:\:MW
two or three days

The annotation for these is not consistent in the Treebank; in some instances, elements \(N_1\) and \(N_2\) adjoin under QP (31-a), but in the majority of cases, these are treated as instances of RNR (31-b).
We demonstrate that the syntax of phrases of estimation is more restricted than is allowed for cases of right node raising, suggesting that the two constructions are distinct:

- \( N_1 \) and \( N_2 \) must differ by one ("sixteen or eighteen"),
- \( N_1 < N_2 \) ("eighteen or seventeen"),
- no more than two numerals may occur, and
- no explicit coordination word may occur between \( N_1 \) and \( N_2 \).

None of these stipulations are placed on right node raising in the general case, so we conclude that estimation phrases are distinct from right node raising. As with the case of coordinator-less \textsc{rnr} described in the previous section, we do not attempt to normalise phrases of estimation in Chinese CCGbank.

### 5.1.6 Argument cluster coordination

Argument cluster coordination (\textsc{acc}) refers to the coordination of syntactic units formed from more than one rightward verbal argument, as in:

(32) Cal gave Jane a green lollipop and Jake a red lollipop.

\(^4\) These constraints do not apply when explicit coordination words appear (e.g. 第十九和第二十和第二十一颗卫星 satellites 19 and 20 and 21) — these are treated instead as conventional coordination.
The PCTB bracketing guidelines contain a facility for marking parallel constructions, of the kind represented by ACC: the coindexation =i on parallel constituents of an argument cluster. For example, the structure of (32) would be represented as:

\[
\text{(33)}
\]

\[
\begin{array}{c}
\text{IP} \\
\text{NP-SBJ} \\
\text{Cal} \\
\text{VP} \\
\text{VV} \\
\text{gave} \\
\text{NP-SBJ}=1 \\
\text{Jane} \\
\text{NP-SBJ}=2 \\
\text{a green lollipop} \\
\text{and} \\
\text{NP-SBJ}=1 \\
\text{Jake} \\
\text{NP-SBJ}=2 \\
\text{a red lollipop}
\end{array}
\]

Note that this encoding violates the endocentricity condition on VP, since it is not headed by a verb. However, the PCTB annotation does not consistently mark ACC instances in this manner. In most cases, instances of ACC have the same structure as (33), but lack the parallel coindexation =i.

As a result, our algorithm identifies ACC instances configurationally, without relying on the parallel coindexation. We define an ACC instance as a VP coordination in which one conjunct (the first) immediately dominates a verbal category, and any conjunct does not immediately dominate a verbal category. This configuration is expressed by the precondition on matching nodes in Algorithm 11.

**Algorithm 11** Re-analysis of argument cluster coordination

Precondition:

1. shrink T
2. re-attach the verb (1) above TOP
3. re-label argument cluster categories

For any two categories AP and BP which form the argument cluster, the re-labelling subroutine in Line 3 of Algorithm 11 type-raises AP and BP so that (< B) obtains the argu-
5.1. Traces

Argument cluster category \( VP\langle\langle VP/BP\rangle/\langle AP \rangle \rangle \). Argument clusters with this category can then be coordinated before consuming the verbal category.

\[
(34) \quad \frac{\langle VP/BP\rangle/\langle AP \rangle \quad \langle AP \rangle \quad \langle BP \rangle \quad \langle BP\rangle\langle \langle VP/BP\rangle/\langle AP \rangle \rangle \quad \langle AP \rangle \quad \langle BP \rangle \quad \langle CP \rangle \quad \langle CP\rangle\langle \langle VP/BP\rangle/\langle AP \rangle \rangle}{\langle VP\rangle\langle \langle VP/BP\rangle/\langle AP \rangle \rangle \quad \langle VP\rangle\langle \langle VP/BP\rangle/\langle AP \rangle \rangle \quad \langle VP\rangle\langle \langle VP/BP\rangle/\langle AP \rangle \rangle}
\]

Because \( \text{PCTB} \) does not contain argument clusters of more than two arguments, we directly implement the above special case (a cluster of two rightward arguments). However, the analysis generalises to arbitrary argument clusters of a bounded size, for instance:

\[
(35) \quad \frac{\langle (VP/CP)/BP\rangle/\langle AP \rangle \quad \langle AP \rangle \quad \langle BP \rangle \quad \langle CP \rangle \quad \langle CP\rangle\langle \langle (VP/CP)/BP\rangle/\langle AP \rangle \rangle}{\langle VP\rangle\langle \langle (VP/CP)/BP\rangle/\langle AP \rangle \rangle \quad \langle VP\rangle\langle \langle (VP/CP)/BP\rangle/\langle AP \rangle \rangle \quad \langle VP\rangle\langle \langle (VP/CP)/BP\rangle/\langle AP \rangle \rangle}
\]

5.1.7 Instances of the 把 \( ba \) and 被 \( bei \) constructions

Table 5.5 presents the \( \text{PCTB} \) annotation for subject- and object-gapped short 被 \( bei \), object-gapped long 被 \( bei \) and the object-gapped 把 \( ba \) construction. All four involve a gap in an \( A \)-position, but the \( \text{CCG} \) analysis of each case differs considerably. We consider each configuration in turn, and describe the re-analysis of each configuration in Chinese CCGbank.
Table 5.5: Three configurations of 被 bei in PCTB and one of 把 ba
5.1. Traces

**Gapped short 被 bei**

We use the following running example to illustrate the re-analysis of subject-gapped short 被bei.

(36) 被认定为高新技术产业的公司
bei recognise be high-tech technology industry de company
a company recognised as (part of the) high-tech industry

In the PCTB annotation for (36), the NP complement of 认定 recognise is a trace, reflecting the subcategorisation frame:

(37) 认定 recognise [ _ ]NP [ _ ]VP

With type-raising and crossed composition, we percolate the NP trace up the derivation:

(38)  a company which was recognised as high-tech industry

To achieve the re-analysis in (38), we shrink the subject trace T, and apply category repair to create the type-raised category. Finally, the category of the particle 被 bei is adjusted so that its argument category is the same as that of pp. The resulting category, together with the head coindexation in Section 3.4.4, allow the non-local dependency to be correctly retrieved.

The details of the analysis of object-gapped short 被 bei (39) are analogous, and we give the re-analysis procedure in Algorithm 13.
Algorithm 12 Re-analysis of subject-gapped short 被 bei
Precondition:

1. shrink T
2. category repair from S to PP
3. BEI.CAT.ARG ← PP.CAT

Algorithm 13 Re-analysis of object-gapped short 被 bei
Precondition:

1. shrink T
2. category repair from S to BEIP
3. BEI.CAT.ARG ← BEIS.CAT

She was accused of murder.

她 被 指控 犯有 间谍罪。

NP VP/TV (VP/VP)/NP TV NP
↓ VP
↓ VP\(\text{\textbackslash}(VP/VP)\)\textsuperscript{T}
↓ TV
↓ VP

S[dl]
S[dl]
5.1. Traces

Object-gapped long 被 bei

While the PCtb bracketing guidelines suggest that the trace in short 被 bei arises from A-movement, it maintains that the object gap in long 被 bei arises from A′-movement instead. This is consistent with the position in Huang et al. (2008), which analyses short 被 bei as involving A-movement and control, and long 被 bei as involving A′-movement and predication. This explains why the gap in short 被 bei is annotated as an A trace *-i, while the gap in long 被 bei is annotated as an A′ trace *T*-i in common with extraction. As such, object-gapped long 被 bei receives the annotation:

(40) underwent inspection by immigration officials

We first re-shape the original annotation so that 被 bei subcategorises for a gapped IP rather than a CP-OBJ complement:

(41)

---

5 The lexical item 所 suo in (40) is a particle that occurs in formal written registers of the relative clause construction. Chiu (1995) analyses 所 suo as being licensed by object extraction.
Algorithm 14 Re-analysis of object-gapped long 被 bei

Precondition:

```
*  
LB IP=S  
VP  
VX=PRED NP-OBJ=T  
*T*-i
```

1. shrink T
2. apply category repair from PRED to S
3. BEI.CAT.ARG ← S.CAT

Then, we shrink the trace *-i and relabel the category of the particle 被 bei, resulting in the CCG analysis:

```
(42) 被 入境处 职员 所 监视
BEI immigration official suo inspect

VP/[S[dcl]] NP/NP NP (S\NP)/(S\NP) TV

NP -> T NP -> B
S/(S\NP) TV -> B

S[dcl]/NP TV -> B
```

Algorithm 14 formalises the procedure, which forms the correct category (S[dcl]\NP)/(S[dcl]/NP) given in Section 3.4.4.
5.1. Traces

**Object-gapped 把 ba**

The PCTB annotation of object-gapped 把 ba is:

(43) encircled us firmly

Recall from Section 3.4.3 that our chosen analysis of the gapped 把 ba construction assigns the particle 把 ba the category \((S[\text{dcl}] \backslash NP) / (S[\text{dcl}] / NP)\). To create the object-gapped clause category \(S[\text{dcl}] / NP\), we find every VP under which an A-trace \(*-i\) lies, shrink the trace, and apply category repair to that subtree. Finally, we relabel the argument category of the particle 把 ba so that it consumes the object-gapped clause category \(S[\text{dcl}] / NP\), resulting in the CCG re-analysis in (44).

\[
\begin{array}{cccccc}
\text{将} & \text{我们} & \text{团团} & \text{包围} & \text{住} \\
BA & 1\text{PL} & \text{gathered} & \text{encircle} & \text{firm:res} \\
\text{VP} / (S[\text{dcl}] / NP) & \text{NP} / (S[\text{dcl}] / NP) & (S[\text{dcl}] / NP) & (S[\text{dcl}] / NP) & (S[\text{dcl}] / NP) \\
S / (S[\text{dcl}] / NP) & (S[\text{dcl}] / NP) & TV & TV & TV \\
S[\text{dcl}] / NP & TV & TV & TV & TV \\
\end{array}
\]
Algorithm 15 Re-analysis of object-gapped 把 ba

Precondition:

\[
\text{TOP} \\
\text{BA} = \text{BA} \\
\ast = \text{C} \\
\vdots \\
\ast - i
\]

1. for every match of PP under N do

\[
\text{NP-OBJ} = \text{T} \\
\text{S} \\
\ast T \ast - i
\]

2. shrink T

3. apply category repair from S to C

4. BA.CAT.ARG \leftarrow C.CAT

5.2 Constructions not handled

Unlike gap extraction

The Coordinate Structure Constraint is a linguistic universal on extraction or movement out of the conjuncts of a coordination.

In a coordinate structure, no conjunct may be moved, nor may any element contained in a conjunct be moved out of that conjunct. (Ross, 1967, p.161)

This constraint is intended to explain the ungrammaticality of extraction out of one conjunct but not another:

(45) ??What, sofa will he put the chair between some table and t??

An exception is available when elements in parallel positions in each conjunct all undergo movement together. This is the Across-the-Board (ATB) condition (Ross, 1967; Williams, 1978), which licenses:

(46) the creature, which John saw t, and Dad fought t,

However, a number of PCtB trees involve coordination of clauses in which the gaps are not all in the same surface argument.
5.2. Constructions not handled

> 0.4(8)

(47) a. which were loaned by international finance organisations and for which bids were publicly solicited from the international community

\[
\text{IP} \quad \text{NP-SBJ} \quad \text{VP} \quad \text{CC} \quad \text{NP-SBJ} \quad \text{VP} \\
\text{international} \quad \text{finance} \quad \text{organisation} \quad \text{loan} \quad \text{and} \quad \text{enter-into} \\
\text{by} \quad \text{public} \quad \text{solicit-bids} \quad \text{international}
\]

> 8.33(5)

b. which French publishers have already published and which have entered into the publishing plan

\[
\text{IP} \quad \text{NP-SBJ} \quad \text{VP} \quad \text{NP-SBJ} \quad \text{VP} \\
\text{France} \quad \text{publisher} \quad \text{already} \quad \text{publish} \quad \text{enter-into} \\
\text{by} \quad \text{international} \quad \text{publish-plan}
\]

In (47-a), a constituent with an object gap (international financial organisations loan \(t_i\)) is coordinated with a constituent with a subject gap (by international public bids). As the common coindexation \(*T*-1\) indicates, both gaps are intended to be coreferent to the same argument. We found nine cases of unlike gap extraction in PCTB.

These appear to be counter-examples to the ATB condition, which contrast with the questionable status of:\(^6\)

(48) a. ??a girl who plays tennis and John knows

b. ??a man who dislikes dogs but cats love

While counter-examples such as What did you go to the store and buy \(t_i\) have been considered in the literature, (Goldsmith, 1985; Lakoff, 1986), we are not aware of any account

\(^6\) (48-b) is marginal to this speaker, while the corresponding phrase with coordinator and seems worse. We are not aware of any accounts of this effect.
of unlike gap coordination in Chinese. Furthermore, since the CCG coordination schema is a categorial analogue of the ATB condition, CCG also lacks a satisfactory explanation of unlike gap extraction phenomena. An analysis through the following rule is possible:

$$S[\text{dcl}] / NP_z \rightarrow S[\text{dcl}] \backslash NP_z \rightarrow S[\text{dcl}] \backslash NP_z$$

but would drastically over-generate. In fact, Steedman (2000) uses the above rule as an example of a rule which contravenes the Principle of Inheritance. The corpus conversion algorithm will generate the above rule, but this is a tentative and unsatisfactory analysis. As such, we do not attempt to extract the expected dependencies from such cases through our gold-standard dependency extraction algorithm.

The investigation of what constitutes acceptable conjuncts to unlike gap extraction is left as an intriguing direction for future work.

**Fragments**

The PCTB phrase-level tag FRAG is used primarily to mark reporter bylines, extra-syntactic material such as strings describing the dimensions of objects, and other specialised syntax such as weather reports and page numbers. Because bylines occur often in PCTB text types, 1932 (6.90%) of trees contain a FRAG. Because of their unevenness, and despite the existence of internal structure, we do not specifically analyse fragment types. In Chinese CCGbank, a phrase annotated as FRAG is mapped to a category $S[frg]$ and treated as right-branching; essentially a non-analysis. This behaviour is consistent with that of English CCGbank (Hockenmaier, 2003), who apply a similar analysis to handle the 611 (1.24%) of PCTB sentences with a FRAG top node.

However, in the special case of a PCTB tree whose root is FRAG and a unary projection of a normal category type, such as PP, we simply delete the FRAG node so that the analysis as the category type can proceed as normal. For instance, this example, which is annotated as a top-level FRAG in PCTB, yields a CCG derivation with the top-level category PP. This condition occurs in 7 (0.03%) of PCTB sentences.

(49)
Pseudo-clefs

We presented two alternative analyses of pseudo-clefs in Section 3.2.4 as part of the abstract CCG grammar — one which retrieves the non-local dependency below, and one which does not.

\[(50) \quad \text{我们需要} \quad t_i \text{的} \quad \text{是} \quad \text{和平} \quad i \text{。} \]
1PL need DE is peace
What we need is peace.

However, the annotation of pseudo-clefs in the PCTB does not support the extraction of the relationship between the gap and the head, because it lacks the coindexation between the clausal gap \( t_i \) and its coreferent NP in the equative construction. As such, we choose the first analysis in Section 3.2.4, which treats the combination \( X + \text{的} \text{de} \) as resulting in type NP, for all pseudo-clefs, regardless of whether it is involved in the equative construction.

V-O construction

The morphologically decomposed analysis of the V-O construction given in Section 2.5.2 is at odds with the PCTB annotation, which treats the full verb as an atomic lexical item, and the cleaved verb as two lexical items:

\[(51) \quad \text{a. 回家} \]
return-home
go home
\[ \text{b. 回了家} \]
return-ASP-home
has gone home

This is despite the fact that the cleaved lexical items are not necessarily free morphemes in the modern language. Because the PCTB annotation does not mark such separable verbs, we do not enact the morphologically decomposed analysis of Section 2.5.2. However, we note that heuristics or manual annotation could be used to recover the distinction, deferring this to future work.

Lexicalised coordination

We concluded in Section 2.6.1 that a lexicalised analysis of coordination would be preferable to the CCGbank-style analysis through coordination type-change rules, on the grounds
that Chinese coordinators, unlike those of English, can bear selectional restrictions on the type of conjuncts they may coordinate. However, such lexical categories of the form $(X \setminus X)/X$ would trigger over-generation in a parser unless the slashes were decorated with the application-only mode ($\star$).

In common with English CCGbank, we chose not to incorporate multi-modal CCG analyses in the concrete version of Chinese CCGbank. As such, the lexicalised coordination analysis without the mode restrictions described in Section 2.6.1 would engender considerable parser ambiguity. As such, we select the type-change analysis of coordination for Chinese CCGbank, in common with English CCGbank.

5.3 Gold-standard dependency extraction

The final stage in the Chinese CCGbank pipeline is the extraction of gold-standard dependencies for evaluation. These gold-standard dependencies are required for the evaluation over labelled dependencies which is standard in the CCG literature (Clark and Hockenmaier, 2002).

We developed an algorithm called MKDEPS, which accepts a CCG derivation and extracts the dependencies implied by the derivation. This makes it similar in operation to the generate tool which the C&C parser uses to induce dependencies from gold-standard derivations in order to extract parser features for training. Nevertheless, we implemented our own algorithm to prevent the incorporation of C&C-specific assumptions.

MKDEPS percolates head-structure from the leaves of a derivation upwards, by applying unification to the matching parts of combinatory rules. Initially, all slots are empty except for the outermost head slot of the lexical items, which are pre-populated with the lexical item itself.

![Diagram](52)

Algorithm MKDEPS re-traces the derivation from the bottom up. First, we consider the forward application between ate and pumpkins, which results in the unifier $y = \text{pumpkins}$.

---

7 The lexicalised coordination constraints can be translated to type-change rules by creating one coordinator category for each possible conjunct type ($\text{conj}_{NP}$, $\text{conj}_{PP}$ and so on). Each coordinator category would be paired with coordination rules which only allowed coordination of that type (for instance, $\text{conj}_{PP} PP \rightarrow PP[conj]$; $PP PP[conj] \rightarrow PP$). We leave the implications of this analysis for future work.
The parent category $S[dlc] \backslash NP$ then gains the coindexation $S[dlc]_{ate} \backslash NP_z$. Then, the backward application between John and the unit ate pumpkins results in the unifier $z = John$, and the algorithm outputs the two dependencies, in the format $(head, category, slot, argument)$:

$(ate, (S[dlc] \backslash NP)/NP, 1, John)$
$(ate, (S[dlc] \backslash NP)/NP, 2, pumpkins)$

One subtlety arises when considering unary type-change rules. Consider the non-gapped topicalisation rule $NP_z \rightarrow [S_y/S_y]_z$. To recover the dependency between 张三 Z. and 喜欢 like below, the algorithm must create the dependency between the variable $y$ in the parent category $S_y/S_y$, and the variable $z$ which holds the head of the $NP$.

We do not reproduce Algorithm $MKDEPS$ in its entirety, as it consists of one branch per combinatory rule which unifies the matching parts of the input categories, and applies the result of the unification to the output category.

### 5.4 Evaluating automatic gold-standard NLD extraction

With a large, automatic corpus conversion effort such as Chinese CCGbank, it is important to validate the resulting analyses in as many ways as possible, to give confidence in the correctness of the generated gold-standard data.

One methodology is to manually create gold standard data, and compare this with the output of the process which generates the silver standard data. The evaluation performed by Clark and Hockenmaier (2002) and other work based on CCGbank compares dependencies returned by the parser against the gold-standard dependencies automatically generated from the data (known as PARG files [predicate-argument annotations] in the CCGbank distribution). While it is not feasible to manually annotate the entire corpus, agreement of the manual gold standard with the automatic silver standard for a section of the data is evidence for the correctness of the silver standard data in the corpus at large.
Annotating gold standard dependencies in a certain way also allows for a fine-grained comparison of parsers. Rimell et al. (2009) argues that a single number, such as labelled $F$-score on dependency tuples, can be misleading, as it is computed over dependencies of all types, disregarding their distribution and the comparative difficulty of recovering them. By choosing types of dependencies known to be challenging to recover, we gain a more detailed breakdown of each parser’s capabilities. We focus on non-local dependencies (NLDS), which Rimell et al. (2009) argue are a stronger test of a parser’s ability than the recovery of brackets. Additionally, an evaluation over NLDS allows the correct interaction of the NLD extraction procedures described in Chapter 5 to be evaluated.

Guo et al. (2007b) have also shown that, comparing the annotations in the Penn Treebank and the Penn Chinese Treebank, Chinese NLDS are both more frequent and occur in a greater variety of species. This suggests that evaluation metrics which do not account for NLDS are even less suitable for evaluating Chinese parsers than they are for English.

Table 5.6 lists the NLD types for which we manually annotated gold-standard dependencies. Compared to the English NLD types annotated by Rimell et al. (2009), we exclude wh-questions (these did not occur sufficiently often in the Penn Chinese Treebank) and free relatives (the corresponding Chinese syntax, 的 de nominalisation, does not involve an NLD); and include the NLDS resulting from gapped topicalisation and the gapped 把 ba and 被 bei constructions.

We undertook a manual annotation task over Section 00 of Chinese CCGbank, in which we automatically detected instances of the NLD types in Table 5.6, and allowed a human annotator\(^8\) to specify the word-word dependencies which should hold between the word projecting the NLD, and the other words in the sentence. The annotator could also specify that the dependency is not filled. We detected instances of each NLD type by writing a \texttt{Tgrep} expression which matches the annotation criteria for each construction as described in Xue et al. (2000). Following Rimell et al. (2009), we developed broadly applicable \texttt{Tgrep} expressions to ensure high recall of instances of each NLD type, then performed manual annotation over the automatically returned candidates to ensure the quality of the resulting dependencies. Unlike Rimell et al. (2009), whose annotations are in the form of Stanford dependencies (de Marneffe et al., 2006), our annotations only consist of unlabelled word-word dependencies together with the type of NLD expected (for instance, (3, 15, subject-extraction)). This is sufficient for the purpose of evaluating the quality of automatic gold-standard dependencies — Rimell et al. (2009) annotates with Stanford dependencies to facilitate multi-parser comparison.

\(^{8}\)In this work, the author annotated the NLD types.
## Type and subtype

<table>
<thead>
<tr>
<th>Type and sub-type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject extraction</td>
<td>overt</td>
</tr>
<tr>
<td></td>
<td>null</td>
</tr>
<tr>
<td>Object extraction</td>
<td>overt</td>
</tr>
<tr>
<td></td>
<td>null</td>
</tr>
<tr>
<td>Gapped topicalisation</td>
<td></td>
</tr>
<tr>
<td>Gapped 把 ba NLDs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>subject</td>
</tr>
<tr>
<td></td>
<td>object</td>
</tr>
<tr>
<td></td>
<td>subject</td>
</tr>
<tr>
<td></td>
<td>object</td>
</tr>
<tr>
<td>Right node raising</td>
<td></td>
</tr>
</tbody>
</table>

| Table 5.6: NLD types for our manual gold-standard annotation |
Chapter 5. Extracting NLD analyses for Chinese CCGbank

In the process of collecting instances of NLD types to annotate manually, we noted that sections 00–09 did not contain any instances of the gapped 把 ba construction at all. Subsequently the construction occurs more frequently in the rest of the corpus, beginning in section 20 (which is also the start of the PCTB 6.0 supplement). This suggests that a partition of the training data should take into account the clumpiness of the PCTB annotation, a concern echoed by Levy and Manning (2003) to justify their non-standard split of an early version of the PCTB, which we previously discussed in Section 4.1.2.

The number of dependencies for which the output of Algorithm MKDEPS matches the manual NLD annotation, and a breakdown of the error cases, is given in Table 5.7. The analysis shows that Algorithm MKDEPS correctly retrieves 504 (90%) of non-local dependencies, which should be the among the most challenging to extract, because of their reliance on heads being passed correctly across unbounded distances. This result quantifies the reliability of the gold-standard dependencies that the algorithm was able to extract.

<table>
<thead>
<tr>
<th>Count</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>583</td>
<td>Total dependencies annotated</td>
</tr>
<tr>
<td>-9</td>
<td>Couldn’t map indices</td>
</tr>
<tr>
<td>-14</td>
<td>Dependency extraction error</td>
</tr>
<tr>
<td>560</td>
<td>Total dependencies which can be compared</td>
</tr>
<tr>
<td>504</td>
<td>Correct dependencies</td>
</tr>
</tbody>
</table>

Table 5.7: Results of manual gold-standard NLD annotation

<table>
<thead>
<tr>
<th>Count</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>Incorrect rules in Chinese CCGbank</td>
</tr>
<tr>
<td>8</td>
<td>Bad PCTB annotation</td>
</tr>
<tr>
<td>5</td>
<td>Rare but legitimate rules unhandled in MKDEPS</td>
</tr>
<tr>
<td>5</td>
<td>Errors in MKDEPS operation</td>
</tr>
<tr>
<td>4</td>
<td>Unlike gap extraction (§5.2)</td>
</tr>
<tr>
<td>4</td>
<td>RNR without coordination word (§5.1.5)</td>
</tr>
</tbody>
</table>

Table 5.8: Error analysis of 56 MKDEPS errors, categorised
The 56 cases for which gold-standard dependencies could not be extracted arise from the unification of filled variables with distinct values, which causes the unification algorithm to fail. We manually categorised the 56 cases into error types, yielding Table 5.8.

This analysis shows that 16 (29%) of errors are caused by incorrect annotation in the PCTB or instances of the unanalysed constructions discussed in Section 5.2, and 10 (18%) are caused by errors in MKDEPS. The remaining 30 cases result from annotation noise yielding incorrect derivations, or constructions not correctly handled by the conversion algorithm. We conclude that MKDEPS itself is responsible for only a small proportion of the errors, and the vast majority of errors can be attributed to other causes such as annotation noise, rare constructions or errors in the conversion algorithm, suggesting that automatic dependency extraction is a viable strategy for inducing gold-standard dependencies from the products of corpus conversion.

Losses during gold-standard dependency extraction

During gold-standard dependency extraction (Algorithm MKDEPS) over the derivations produced by the Chinese CCGbank algorithm, 151 (0.54%) fail the extraction process. For these sentences, Chinese CCGbank contains their derivation trees, but not their gold-standard dependencies. Out of these, 88 (58%) fail because the conversion process has resulted in invalid CC rules. These invalid rules arise from legitimate but rare constructions such as unlike coordinated phrases (UCP in the PCTB annotation) and coordination of unlike gaps (Section 5.2), as well as limitations of the conversion algorithm.

5.5 Analysis of Chinese CCGbank

Hockenmaier and Steedman (2002b) showed that in formalisms which use rich tagsets like ccg, the impact of sparsity is compounded by the number of categories in the lexicon. We analyse properties of Chinese CCGbank, focusing on sparsity and the ability of the conversion algorithm to discharge the traces which mark NLDS in the PCTB annotation and which CCG disallows.

5.5.1 The lexicon

Despite the lexicon being smaller in size (due to the lack in Chinese of morphological variants marked through features) compared to the English CCGbank lexicon, the Chinese CCGbank lexicon demonstrates a similar degree of lexical ambiguity. The twenty lexical
items with the greatest lexical ambiguity are given in Table 5.9. A similar table for English CCGbank can be found in Hockenmaier (2003), in which the twenty lexical items with greatest lexical ambiguity account for 1327 categories. In Chinese CCGbank, the top twenty lexical items by lexical ambiguity account for 1256 categories.

We examined the categories arising from the high-frequency function word 的 de, which has 182 attested categories in Chinese CCGbank. The high degree of lexical ambiguity is due to two interlocking factors:

的 de is polysemous. 的 de is the Chinese relativiser (Section 3.2.1), occurring in both gapped and non-gapped frames; the generalised modification particle (Section 3.2.3); and the nominaliser particle and a part of the pseudo-cleft construction (Section 3.2.4). Each sense of 的 de requires a distinct shape of category.

X de Y is highly productive. In generalised modification (Section 3.2.3), one 的 de category results from each combination of X and Y, leading to lexical ambiguity. In the relative clause construction, one 的 de category results from each possible attachment level of the relative clause.

The cause of the lexical ambiguity encountered for 的 de (180 categories) is similar to that of the most lexically ambiguous English CCGbank entry as (134 categories). The lexical ambiguity of as also arises from its ability to join many syntactic types (coming as it does, such as real estate, as if to say), although its degree of polysemy does not match that of 的 de.

<table>
<thead>
<tr>
<th># cats</th>
<th>Frequency</th>
<th>Token</th>
<th># cats</th>
<th>Frequency</th>
<th>Token</th>
</tr>
</thead>
<tbody>
<tr>
<td>182</td>
<td>39394</td>
<td>的 de</td>
<td>50</td>
<td>1280</td>
<td>对 for</td>
</tr>
<tr>
<td>99</td>
<td>9924</td>
<td>在 at</td>
<td>49</td>
<td>934</td>
<td>从 from</td>
</tr>
<tr>
<td>80</td>
<td>7832</td>
<td>是 is</td>
<td>49</td>
<td>2852</td>
<td>前 before</td>
</tr>
<tr>
<td>79</td>
<td>6200</td>
<td>一 one</td>
<td>48</td>
<td>1607</td>
<td>来 come</td>
</tr>
<tr>
<td>68</td>
<td>943</td>
<td>（</td>
<td>47</td>
<td>1689</td>
<td>等 etc.</td>
</tr>
<tr>
<td>68</td>
<td>1900</td>
<td>到 to</td>
<td>46</td>
<td>1926</td>
<td>以 with</td>
</tr>
<tr>
<td>60</td>
<td>3884</td>
<td>有 have</td>
<td>45</td>
<td>3492</td>
<td>这 this</td>
</tr>
<tr>
<td>55</td>
<td>2291</td>
<td>上 above</td>
<td>44</td>
<td>719</td>
<td>下 under</td>
</tr>
<tr>
<td>51</td>
<td>6403</td>
<td>了 ASP</td>
<td>43</td>
<td>2386</td>
<td>中 middle</td>
</tr>
<tr>
<td>51</td>
<td>2458</td>
<td>为 for</td>
<td>42</td>
<td>942</td>
<td>三 three</td>
</tr>
</tbody>
</table>

Table 5.9: Most ambiguous lexical items in Chinese CCGbank
### Analysis of Chinese CCGbank

#### Frequency Type

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Right node raising without CC</td>
</tr>
<tr>
<td>5</td>
<td>Wrong attachment of DEC in RC</td>
</tr>
<tr>
<td>5</td>
<td>Conversion fails on right node raising</td>
</tr>
<tr>
<td>4</td>
<td>Incorrect non-gapped topicalisation annotation</td>
</tr>
<tr>
<td>3</td>
<td>Incorrect right node raising annotation</td>
</tr>
<tr>
<td>3</td>
<td>Other bad annotation</td>
</tr>
</tbody>
</table>

Table 5.10: Error analysis of coverage losses

#### Coverage losses

Chinese CCGbank yields analyses for 27944 (98.76%) of PCTB trees. Derivations may be filtered out in the pipeline due to annotation noise, constructions not covered by Chinese CCGbank, or unexpected interactions between the PCTB annotation and the conversion algorithm.

From the initial PCTB corpus of 28295 trees, no trees are lost by the marking algorithm (Algorithm 3), binarisation, or category labelling (Algorithm 4), but 35 trees are lost during trace elimination (Section 4.6.5). Table 5.10 decomposes the 35 trees lost through the conversion pipeline by cause of coverage loss.² Twenty cases are due to an unexpected right node raising annotation, where no coordination word separates the conjuncts. In some of these cases, RNR is used to analyse the bare coordination of verbs, which usually receives VCD in the PCTB annotation. Our algorithm fails on these cases, as we assume that a coordination word is always present.

We then filter out trees for which the conversion algorithm has failed to eliminate all traces, and trees which contain atoms outside the allowed set. The latter error condition is often caused by incorrect head structure annotations in the corpus. This results in a final corpus size of 27944 (98.76%) derivations.

The corpus also contains 1333 trees which consist of the text (完) (end) or reporter bylines, which may not be useful in many applications. Removing these yields a corpus of 26611 derivations, which we use for the parsing experiments described in the following pair of chapters.

---

² Out of the five “wrong attachment of DEC” cases, four occur close to one another in the numeric sequence of PCTB documents.
5.5.3 The effect of sparsity

Lexical item coverage quantifies the impact of sparsity due to unseen words. Hockenmaier and Steedman (2002b) showed that formalisms with rich tagsets, such as CCG, are particularly sensitive to this sparsity — while a lexical item may be attested in the training data, it may lack the necessary category.

Hockenmaier and Steedman (2007) examines the category growth behaviour of English CCGbank to determine the likelihood that the English CCGbank lexicon is complete. They incrementally uncover sentences from the corpus and count the number of seen category types and rule instantiations at each stage. If, close to the true size of the corpus, adding more sentences does not greatly increase the lexicon size or number of rule instantiations, we can conclude that the corpus contains nearly all the categories that the true lexicon for English requires.

For Chinese CCGbank, Figures 8.7a and 8.7c show that once a frequency threshold of 5 is applied, neither lexicon size nor number of rule instantiations grows substantially past 600000 tokens. Figures 8.7b and 8.7d show that the frequencies of categories and rule instantiations roughly follow the expected power law distribution.

We divided the tokens of 27944 valid derivations into ten folds, performing ten-fold cross-validation to determine the coverage of lexical items and CCG categories in the resulting corpus. As demonstrated in Section 4.1.1, the Penn Chinese Treebank is more heterogeneous than the Penn Treebank. We provide two coverage figures to compare the CCG-banks derived from Chinese and English Penn Treebank: one with random assignment of derivations to folds, and one with folds taken over contiguous segments of the corpus.

Table 5.11 shows that sparsity both over lexical items and categories is greater in Chinese CCGbank compared to English.

5.5.4 Discharging NLD traces

One of the challenges of developing corpus conversions from formalisms based on movement and traces to monosstral trace-free formalisms, is reshaping the source analyses into their trace-free counterparts. We evaluate how well the conversion algorithm described in this chapter discharges the traces which arise from the analysis of various NLD constructions in the PCTB annotation style.

For each sentence in Chinese CCGbank, we aligned the leaves of the original PCTB annotation (which may contain traces) to the leaves of the corresponding Chinese CCGbank derivation. We then found instances of a number of NLD types, using the NLD finding ex-
5.5. Analysis of Chinese CCGbank

Figure 5.4: Category growth behaviour on the Penn Chinese Treebank
expressions presented earlier in Section 5.4 to find manual annotation instances for each NLD type, and judged the algorithm to have discharged a given trace, if a trace in the source annotation was not aligned to any leaf in the target annotation. The proportion of NLD traces of each type discharged by the algorithm is given in Table 5.12.

The coverage of right node raising (RNR) is the lowest of the NLD types, due to the presence of estimation phrases and bare verb coordination annotated as RNR. As we argued in Section 5.1.5, these are not cases of right node raising at all. The next most frequently unanalysed NLD type is extraction; these are instances of the unlike gap extraction construction discussed in Section 5.2. To our knowledge, no formalism in which an analogue of the Coordinate Structure Constraint applies to constrain the types of conjuncts has an account for this construction, which our algorithm likewise does not handle.

<table>
<thead>
<tr>
<th></th>
<th>random by type</th>
<th>random by token</th>
<th>contiguous by type</th>
<th>contiguous by token</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lexical items</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chinese</td>
<td>83.16%</td>
<td>96.85%</td>
<td>74.92%</td>
<td>94.52%</td>
</tr>
<tr>
<td>English</td>
<td>85.37%</td>
<td>97.98%</td>
<td>81.76%</td>
<td>97.17%</td>
</tr>
<tr>
<td>German</td>
<td>92.0% (by token)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Categories</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chinese</td>
<td>89.78%</td>
<td>99.94%</td>
<td>88.16%</td>
<td>99.92%</td>
</tr>
<tr>
<td>English</td>
<td>92.44%</td>
<td>99.96%</td>
<td>91.93%</td>
<td>99.95%</td>
</tr>
<tr>
<td>German</td>
<td>86.7% (by token)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.11: Average lexical and category coverage across CCGbanks (ten-fold cross-validation)
### Summary

This chapter presented the creation of Chinese CCGbank, the first corpus of Chinese CCG derivations in the literature, embodying the new analysis of Chinese syntax described in the previous two chapters. We presented a corpus conversion procedure, adapting the conversion pipeline of Hockenmaier (2003) to obtain a corpus of almost 750,000 words and 26,611 sentences, converted automatically from Penn Chinese Treebank trees.

We have shown that all five annotated NLD types in the Penn Chinese Treebank — pro-drop, PRO, A-movement, A’-movement and right node raising — can be reshaped into equivalent CCG analyses, using a flexible approach which pattern-matches sub-structures of Penn Chinese Treebank trees using Tgrep-like expressions so that they can be reshaped.

We have shown that the gold-standard head structure annotations in the PCTB are suitable for corpus conversion, finally validating the hope expressed in Xue et al. (2005) that providing gold-standard head structure would allow for the creation of high-quality corpus conversions. These gold-standard annotations thus eliminate a source of noise common to corpus conversion procedures — the use of heuristics to identify head structure.

We analysed the sparsity characteristics of the resulting corpus, following the observation by Hockenmaier and Steedman (2002a) that lexical coverage of a corpus affects the performance of parsers trained on it, discovering that the growth rate of novel category

---

<table>
<thead>
<tr>
<th>NLD type</th>
<th>Proportion of traces discharged</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject extraction</td>
<td>10775/10802 (99.8%)</td>
</tr>
<tr>
<td>Object extraction</td>
<td>3105/3114 (99.7%)</td>
</tr>
<tr>
<td>Null-relativiser subject extraction</td>
<td>916/917 (99.8%)</td>
</tr>
<tr>
<td>Gapped topicalisation</td>
<td>725/731 (99.2%)</td>
</tr>
<tr>
<td>Right node raising</td>
<td>570/599 (95.2%)</td>
</tr>
<tr>
<td>Null-relativiser object extraction</td>
<td>535/537 (99.6%)</td>
</tr>
<tr>
<td>Object-gapped short 被 bei</td>
<td>383/384 (99.7%)</td>
</tr>
<tr>
<td>把 ba construction</td>
<td>182/183 (99.5%)</td>
</tr>
<tr>
<td>Object-gapped long 被 bei</td>
<td>167/167 (100.0%)</td>
</tr>
<tr>
<td>Subject-gapped short 被 bei</td>
<td>11/11 (100.0%)</td>
</tr>
<tr>
<td>Subject-gapped long 被 bei</td>
<td>4/4 (100.0%)</td>
</tr>
</tbody>
</table>

Table 5.12: Traces discharged by the conversion algorithm, by NLD type
types and rule types becomes small at approximately 90% of the total corpus size. This demonstrates that despite being only 70% the size of the Penn Treebank, the Penn Chinese Treebank is still sufficiently large that a usable wide-coverage corpus can be extracted from it.

We have demonstrated that the abstract CCG analyses of Chapters 2 and 3 can largely be reified through corpus conversion from a wide-coverage corpus, identifying exceptional syntax for which CCG has no satisfactory analysis, and analyses which the PCTB annotation would not support without further processing.

We have shown that the encoding of Chinese syntax can be extracted automatically from the PCTB annotation, and that the static characteristics of the lexicon are suitable for parsing. We are now ready to perform parsing experiments on the newly created Chinese CCGbank, obtaining the first statistical models of Chinese from a CCG grammar.
Chapter 6

Parsing Chinese CCGbank

This chapter presents the first experiments in Chinese ccg parsing using Chinese CCGbank, with the goal of establishing the utility of Chinese CCGbank for wide-coverage parsing, and quantifying the difficulty of reproducing Chinese ccg analyses in a parser.

A gap separates the state-of-the-art results in phrase structure parsing (Zhang and Clark, 2009) and English psg parsing (McClosky et al., 2006). In this chapter, we investigate whether a similar gap can be observed in Chinese ccg parsing through a multi-parser evaluation.

With Chinese CCGbank, we train three wide-coverage parsers, all of which have achieved state-of-the-art parsing accuracy. We follow three experiments which all train wide-coverage ccg parsers (Clark and Curran, 2007b; Fowler and Penn, 2010; Zhang and Clark, 2011). We not only discover that a large gap exists between the Chinese and English CCGbank parsing results, but also that the three parsers, whose results on English CCGbank were within 0.5% of each other, vary dramatically on Chinese CCGbank.

Chinese CCGbank has shown that not only is a ccg account of Chinese syntax possible, it can be extracted automatically from the pctb. This chapter validates the thesis of this work, showing that Chinese ccg parsing with Chinese CCGbank is also possible. However, it also uncovers surprising gaps in parser performance prompting further investigation, and leaves substantial challenges for future research in Chinese ccg parsing.
Material in this and the following chapter was presented at the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) in Montréal, Canada (Tse and Curran, 2012).
6.1 A roadmap

The variety of experiments in this chapter, and the exposition required to discuss the parsers under test and the experiments performed on them, prompted us to provide a summary similar to the roadmap in the introductory chapter of this thesis.

We begin by surveying approaches in the literature to Chinese parsing. The fact that techniques achieving the state-of-the-art in one language do not necessarily achieve it in the other foreshadows our own observations later in the chapter.

Since the publication of English CCG-bank by Hockenmaier (2003), the literature has been enriched by a body of work describing various mechanisms for CCG parsing. We survey the approaches which have been proposed and implemented, including the three parsers under test in this work.

Of the three CCG parsers under test, two of them rely on components of the first parser, C&C, to produce dependencies in the format expected by the standard dependency-based CCG evaluation, and to perform the evaluation itself. Since these components are shared by all three systems, we describe their operation and the implications on parsing.

We then present the three parsers with which we acquire the first statistical parsing models inferred from Chinese CCGbank: the Clark and Curran (C&C; 2007b) parser, the transition-based parser of Zhang and Clark (Z&C; 2008), and the split-merge PCFG
Chapter 6. Parsing Chinese CCGbank

parser of Petrov and Klein (P&K; 2007), repurposed for ccg parsing by Fowler and Penn (2010). All three parsers differ considerably with respect to inference and decoding method, efficiency, the data structure used to hold partial parses, and their native formalism. We describe the theoretical basis for each parser, the way each parser operates, its training regimen including tuneable parameters, and summarise previous experiments performed with each.

Having described the three parsers, we present our empirical results for Chinese ccg parsing on Chinese CCGbank, discovering that while only a small gap separated the results for the P&K and C&C parsers on English data (Fowler and Penn, 2010), a surprisingly large gap separates the corresponding results on Chinese.

We then describe an experiment to vary the annotation decisions in Chinese CCGbank, observing the impact of these changes on parser models trained on the corpus variants. This experiment also raises the possibility of application-specific corpora which enact a range of annotation decisions.

However, despite the fact that particular annotation decisions which trade lexical ambiguity for derivational ambiguity increase performance on both C&C and P&K, the performance gap between Chinese and English parsing results remains.

To understand the causes of the performance deficit, we describe a number of experiments which quantify the role of tagger error and the impact of coverage loss on different parsers, and focus on the behaviour of each parser in retrieving the more difficult non-local dependencies for which we crafted ccg analyses in Chapter 3.

6.2 A survey of approaches to Chinese statistical parsing

The stature of the Penn Treebank in English statistical parsing, persisting through to the present day, is indicative of the importance of a large, wide-coverage, annotated corpus to statistical parsing research. Similarly, it was only with the introduction of the Penn Chinese Treebank (Xue et al., 2005) that work in wide-coverage Chinese statistical parsing began to flourish.

Bikel and Chiang (2000) applied parsing models developed for English to the newly available Penn Chinese Treebank, determining that Chinese was sufficiently similar to English for these models to yield basic results without having to adapt the models greatly. This conclusion is not surprising, given that Chinese and English both exhibit canonically configurational syntax, the same canonical word order of svo and the same headedness parameters in NP and VPs. Consequently, the major approaches to Chinese statistical parsing
in the last decade have operated on the idea that to tackle Chinese parsing, we can at least begin with models, formalisms and techniques developed for and established in English.

However, Levy and Manning (2003) established that the optimistic idea that the bottleneck observed by Bikel and Chiang (2000) between Chinese and English parsing performance was largely due not to fundamental differences between the languages, but to the relative dearth of training data, was not an adequate characterisation. They showed that Chinese PCFG parsing exhibits kinds of errors not encountered processing English text, and that Chinese parsing models benefit from incorporating features particular to Chinese grammar. In Levy and Manning (2003), many of the familiar characteristics of Chinese grammar which complicate parsing were identified, including noun/verb categorial fluidity, ambiguity arising from pro-drop, and low morphology.

Both Bikel and Chiang (2000) and Levy and Manning (2003) operate on gold-standard Chinese segmented words. This introduces an additional pre-processing stage (and a potential source of error) which English does not require. While the assumption of gold-standard input is common in a number of NLP tasks (for instance, gold-standard POS tags for parsing) where high-accuracy systems for the pre-processing task exist, Chinese word segmentation is not yet considered a solved task.

A number of papers, including Duan et al. (2007) and Zhang and Clark (2008), successfully applied transition-based shift-reduce parsing approaches to Chinese. In the setting of transition-based statistical parsing, a parse tree is incrementally constructed by a statistical model which selects a parse action given a stack of forests and a queue of input tokens.

Zhang and Clark (2009) observes that while constituent-based parsers outperformed transition-based parsers in English, the converse was true of the state of Chinese parsing at the time. To our knowledge, no work has examined what properties of Chinese syntax might be responsible for this effect.

Many statistical parsers for English which report state-of-the-art results on Parseval, including the commonly used Collins (1999) and Charniak and Johnson (2005) parsers, do not recover the annotations which mark non-local dependencies such as those arising from extraction and wh-questions.

Guo et al. (2007b) argue that recovery of non-local dependencies is even more pressing in Chinese parsing, showing that the Penn Chinese Treebank annotation distinguishes more NLD types than does the Penn Treebank, and that these NLD types occur at a higher rate in Chinese sentences than in English. They develop an LFG translation of the Penn Chinese Treebank which converts the traces which mark non-local dependencies in the Penn Chinese Treebank annotation into a structure-sharing LFG representation, and a pro-
cEDURE which restores the re-entrancies which mark NLDs in LFG to the $f$-structures output by an LFG parser. Guo et al. (2007b) observed that while their algorithm performed well recovering NLDs given gold PCFG $f$-structures with the re-entrancies removed, sensitivity to noise and mis-analysis in automatic trees led to a decrease in performance of $\sim 30\%$. This compared to a drop of $\sim 7 - 9\%$ observed by Johnson (2002) when moving from gold to automatic trees on a trace recovery post-processing task, which Guo et al. (2007b) attribute to the increased difficulty of parsing Chinese.

Guo et al. (2007b) highlights the fact that Chinese parsing without NLD recovery greatly underestimates the difficulty of the task, and that a post-processing approach for NLD recovery is highly sensitive to the quality of parser output. This conclusion was the impetus for this dissertation, in order to determine whether CCG, a formalism capable of transparent, accurate NLD recovery in English, could yield similar benefits for Chinese.

### 6.3 Approaches to CCG parsing

The question of how to parse categorial grammar arose as early as Lambek (1958), despite his focus on categorial grammar as a formal description language rather than a basis for computational syntax. Lambek suggested a decision procedure exponential in the length of the string to determine whether it could derive a given top-level category in a given categorial grammar: non-deterministically choose an assignment of categories to the tokens, assign all binary bracketings to the string, apply all available reductions, and answer whether any of the reductions yield the desired top-level category.

Steedman (2000) observed that the binary-branching nature of CCG combinatory rules allow two classes of bottom-up parsers: shift-reduce parsing, and the CKY algorithm, to be adapted easily to CCG parsing. Out of these two, CKY chart parsing approaches predominate in the literature.

Clark et al. (2002) developed an early model for CCG parsing computed over dependencies, and Hockenmaier (2003) developed a statistical model for head-driven generative parsing in the vein of Collins (1999), also beginning with an unlexicalised model and gradually increasing the degree of lexicalisation.

Clark and Curran (2003) first applied a discriminative model to CCG parsing, arguing for the flexibility of log-linear models to incorporate arbitrary, overlapping features of the candidate derivation into the inference process. They use the feature forest method of Miyao and Tsujii (2002) to efficiently compute features over all derivations in a packed
chart, and incorporate a model over the dependency tuples in the candidate derivation. This line of research developed into the Clark and Curran (C&C) parser.

Fowler and Penn (2010) showed that the English CCGbank grammar, considered only as the fixed set of rule instantiations attested in the data, can be interpreted as a context-free grammar where the set of productions are the set of CCG rules in the data. As such, PCFG parsers could be trained directly on English CCGbank, which Fowler and Penn (2010) test by applying the state-of-the-art split-merge CFG parser of Petrov and Klein (2007, P&K) to the CCG corpus. This allowed them to obtain a model with performance competitive with the best C&C models on English, using a parser not specifically developed for CCG.

More recently, Zhang and Clark (2011, Z&C) were the first to implement the second class of bottom-up parsers suggested by Steedman (2000) as a natural match for CCG parsing, implementing a wide-coverage shift-reduce parser with parse actions pruned with a narrow beam. They achieved performance comparable to the CKY-based C&C parser.

We focus on the last three methods of CCG parsing just presented, as these are the three parsers with which we test the effectiveness of Chinese CCGbank for acquiring wide-coverage parsing models of Chinese syntax.

6.3.1 C&C: log-linear models for efficient CCG parsing

The Clark and Curran (2007b, C&C) parser consists of a cascade of tagging and parsing models operating in a pipeline. Clark (2002) first used a supertagger in CCG parsing, using contextual predicates which had access to POS tags to obtain a maximum-entropy model, and this supertagging approach was subsequently integrated with a CCG parser in Clark and Curran (2004b). Clark and Curran (2007b) showed that a limited degree of interaction between the tagging and parsing models, through the technique of adaptive supertagging, allows C&C to balance efficiency, coverage and accuracy. Recall that CCG parsing begins with the assignment of categories to the input lexical items. In C&C, this is performed by applying the supertagger model to POS tagged data. To exercise the trade-off between coverage and efficiency, Clark and Curran (2007b) uses multitagging to initially assign a set of high-confidence categories to each input lexical item, controlling the amount of lexical ambiguity through a parameter $\beta$. Candidate categories whose score is within a factor of $\beta$ of the highest-scoring category will be returned by the multitagger. Thus, the effect of decreasing $\beta$ has the effect of increasing ambiguity, since the ratio between the best-scoring

---

1 Clark and Curran (2010) is a summary of the work in CCG supertagging.
category’s probability and the candidate category’s probability is permitted to grow large. Figure 6.1 shows the impact of $\beta$ on supertagger ambiguity, showing that as $\beta$ decreases, further categories may be proposed for each word.

The input categories are loaded into the chart in preparation for parsing. If the parser fails to find a spanning analysis given the set of categories assigned, adaptive supertagging increases the degree of lexical ambiguity by reducing $\beta$, and parsing occurs again on an expanded set of categories. The process is continued at a number of ambiguity ($\beta$) levels until a spanning analysis is returned by the parser; otherwise, parsing fails. This approach yields higher coverage, because analyses which require rarer categories should receive those categories at higher ambiguity levels, while maintaining efficiency, by withholding those rarer categories until an analysis is impossible given a less ambiguous category assignment. Surprisingly, however, Clark and Curran (2004b) discovered that adaptive supertagging even increases accuracy relative to a configuration in which the supertagger uses only one high-ambiguity $\beta$ level, by withholding low-confidence categories until they are necessary for an analysis.

The C&C POS tagger and supertagger share an architecture (Curran and Clark, 2003). Both are maximum entropy taggers after Ratnaparkhi (1996), extracting local contextual predicates as features and training a maximum entropy model on these overlapping features. A maximum entropy model has the form:

$$P(y|x) = \frac{1}{Z(x)} \exp(\lambda \cdot f(x, y))$$

(6.1)

where $x$ is the candidate class, $y$ is the observed instance, $\lambda$ is a weight vector, $f(x, y)$ is the feature vector extracted from the candidate class and the instance, and $Z(x)$ is a partition function to ensure a probability distribution.

The parameters of the maximum entropy model, $\lambda$, can be inferred from data using any of a number of optimisation methods (Malouf, 2002), but C&C uses Generalised Iterative Scaling (Darroch and Ratcliff, 1972, GIS) or BFGS optimisation (Nocedal and Wright, 2006). The Viterbi algorithm finds the most probable tag sequence given this model.

The stock C&C POS tagger operates in 1-best mode, although Curran et al. (2006) also investigated multi-POS tagging, demonstrating that maintaining a degree of POS ambiguity improves accuracy in the subsequent supertagger phase in English.

Clark and Curran (2007b) present three parser models, which extract parse features either from derivation structure, or the dependencies the candidate derivation yields; the choice of model also affects how decoding is performed. A summary of the three models is given in Table 6.1.
### Figure 6.1: The impact of $\beta$ on multitagger ambiguity

<table>
<thead>
<tr>
<th>Gloss</th>
<th>POS</th>
<th>$\beta = 1$</th>
<th>$\beta = 0.1$</th>
<th>$\beta = 0.05$</th>
<th>$\beta = 0.01$</th>
<th>$\beta = 0.005$</th>
</tr>
</thead>
<tbody>
<tr>
<td>他 3sg</td>
<td>PN</td>
<td>$NP$</td>
<td>$(VP[dcl]/VP[dcl])/NP$</td>
<td>$NP/NP$</td>
<td>$(S[dcl]/S[dcl])/VP[dcl]$</td>
<td>$VP[dcl]/NP, VP[dcl]/VP[dcl]$</td>
</tr>
<tr>
<td>预计 estimate</td>
<td>VV</td>
<td>$NP$</td>
<td>$VP[dcl]/S[dcl]$</td>
<td>$S/S$</td>
<td>$VP[dcl]/VP[dcl]$</td>
<td>$(VP/VP)/(VP/VP)$</td>
</tr>
<tr>
<td>明年 next year</td>
<td>NT</td>
<td>$NP$</td>
<td>$VP[dcl]/VP[dcl]$</td>
<td>$VP/VP$</td>
<td>$VP[dcl]$</td>
<td>$VP[dcl]/VP[dcl]$</td>
</tr>
<tr>
<td>银行业 banking</td>
<td>NN</td>
<td>$NP$</td>
<td>$VP[dcl]/VP[dcl]$</td>
<td>$VP/VP$</td>
<td>$VP[dcl]$</td>
<td>$VP[dcl]/VP[dcl]$</td>
</tr>
<tr>
<td>会 will</td>
<td>VV</td>
<td>$NP$</td>
<td>$VP[dcl]/VP[dcl]$</td>
<td>$VP/VP$</td>
<td>$VP[dcl]$</td>
<td>$VP[dcl]/VP[dcl]$</td>
</tr>
<tr>
<td>更加 additionally</td>
<td>AD</td>
<td>$NP$</td>
<td>$VP[dcl]/VP[dcl]$</td>
<td>$VP/VP$</td>
<td>$VP[dcl]$</td>
<td>$VP[dcl]/VP[dcl]$</td>
</tr>
<tr>
<td>困难 be troubled</td>
<td>VA</td>
<td>$NP$</td>
<td>$VP[dcl]/VP[dcl]$</td>
<td>$VP/VP$</td>
<td>$VP[dcl]$</td>
<td>$VP[dcl]/VP[dcl]$</td>
</tr>
<tr>
<td>。</td>
<td>PU</td>
<td>$NP$</td>
<td>$VP[dcl]/VP[dcl]$</td>
<td>$VP/VP$</td>
<td>$VP[dcl]$</td>
<td>$VP[dcl]/VP[dcl]$</td>
</tr>
</tbody>
</table>
Chapter 6. Parsing Chinese CCGbank

POS tagger

Supertagger

Parser

Adaptive supertagging

Figure 6.2: Architecture of the C&C parser

<table>
<thead>
<tr>
<th>Model</th>
<th>Input data</th>
<th>Decoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal-form</td>
<td>CCGbank derivations</td>
<td>Viterbi algorithm over packed chart</td>
</tr>
<tr>
<td>Dependency</td>
<td>Gold-standard dependencies obtained by running C&amp;C in generate mode over CCGbank derivations</td>
<td>Derivation which maximises expected labelled recall over candidate dependencies</td>
</tr>
<tr>
<td>Hybrid</td>
<td>As in dependency model, but only normal-form derivations allowed in chart</td>
<td>As in dependency model</td>
</tr>
</tbody>
</table>

Table 6.1: Parser models in Clark and Curran (2007b)
During parser training, features are extracted from positive examples (analyses which yield the gold-standard derivation structure or dependency structure, depending on the parser model chosen), and negative examples (analyses which do not). Only features which appear in at least one positive example are used in the model. Parser training is based on the limited memory BFGS (L-BFGS) algorithm (Nocedal, 1980), for which Clark and Curran (2003) describe a parallelised variant which distributes the memory requirements across a cluster of machines.

Of the three parser models in Table 6.1, the highest accuracy for labelled dependency recall on PTB section 00 was achieved with the hybrid model, as Clark and Curran (2007b) found for English.

Table 6.2 summarises the critical parameters for the various components in the C&C pipeline. The Gaussian smoothing parameters and the number of GIS iterations for POS tagging were optimised by grid search on the Chinese CCGbank development set. The $\beta$ sequences of ambiguity levels for adaptive supertagging were determined empirically. We note that the $\beta$ sequence required to achieve acceptable coverage in Chinese involves higher ambiguity levels than the default used in English C&C models, which for the best-performing hybrid model is $\langle 0.0045, 0.0055, 0.01, 0.05, 0.1 \rangle$ for training, while the sequence $\langle 0.075, 0.03, 0.01, 0.005, 0.001 \rangle$ is used during testing. We raised the maximum size of the parser chart (maximum supercats) from the 1 000 000 reported in Clark and Curran (2007b) to 5 000 000. This value reflects the additional ambiguity inherent in Chinese parsing, and was selected empirically to yield acceptable coverage on the development data.

### 6.3.2 P&K: PCFG-based CCG parsing

Following Klein and Manning (2003), who manually introduced linguistically-motivated non-terminal splits in a PCFG, Petrov et al. (2006) developed a PCFG inference method which learns rule probabilities given a particular split of the non-terminals, and show that the decrease in likelihood when a given split is removed can be used to judge the effectiveness of a given candidate split. Performed iteratively, split-merge PCFG inference learns a number of fine syntactic and semantic distinctions automatically, while avoiding the cost of manual annotation.

Fowler and Penn (2010) observed that a parser with the “seen rules” restriction — one which can only produce the CCG rule instantiations attested in the training data — is equivalent to a context-free grammar which simply contains the attested rules as the production rules. This suggested that a state-of-the-art PCFG parser could be trained directly on
### Table 6.2: C&C parameters (chosen values in parentheses)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>POS tagger maxent parameters</strong></td>
<td></td>
</tr>
<tr>
<td>sigma</td>
<td>Parameter of Gaussian smoothing in gis (1.414)</td>
</tr>
<tr>
<td>niterations</td>
<td>Number of gis iterations (350)</td>
</tr>
<tr>
<td><strong>POS tagger features</strong></td>
<td></td>
</tr>
<tr>
<td>prosodic_features</td>
<td>Enables prosodic features</td>
</tr>
<tr>
<td>reduplication_features</td>
<td>Enables reduplication features</td>
</tr>
<tr>
<td><strong>Supertagger window features</strong></td>
<td></td>
</tr>
<tr>
<td>biwords/bitags</td>
<td>bigrams of words/tags in a window up to two tokens</td>
</tr>
<tr>
<td>biwords_far/bitags_far</td>
<td>bigrams of words/tags in a window up to three tokens</td>
</tr>
<tr>
<td>triwords/tritags</td>
<td>trigrams of words/tags in a window up to two tokens</td>
</tr>
<tr>
<td>triwords_far/tritags_far</td>
<td>trigrams of words/tags in a window up to three tokens</td>
</tr>
<tr>
<td>surr3words/surr3tags</td>
<td>unigrams of words/tags three tags in front/behind</td>
</tr>
<tr>
<td><strong>Parser adaptive supertagging parameters</strong></td>
<td></td>
</tr>
<tr>
<td>Training (\beta) sequence</td>
<td>The supertagger ambiguity levels during training ((0.055, 0.01, 0.05, 0.1))</td>
</tr>
<tr>
<td>Parsing (\beta) sequence</td>
<td>The supertagger ambiguity levels during parsing ((0.15, 0.075, 0.03, 0.01, 0.005, 0.001))</td>
</tr>
<tr>
<td>Tagdict thresholds</td>
<td>Thresholds for consulting the tagdict</td>
</tr>
<tr>
<td><strong>Parser parameters</strong></td>
<td></td>
</tr>
<tr>
<td>Maximum supercats</td>
<td>The maximum number of supercats (category entries) in the chart ((5,000,000))</td>
</tr>
</tbody>
</table>

*Chapter 6. Parsing Chinese CCGbank*
English CCGbank without modification, and the resulting model easily compared against other CCG parsers trained on the same data.

They showed that training on English CCGbank, a PCFG-based CCG model can be acquired with the parser of Petrov and Klein whose performance is comparable with the hybrid model, the best C&C model reported by Clark and Curran (2007b). This result was the first to demonstrate that accurate CCG parsing was possible without a dedicated CCG parser.

Following Fowler and Penn (2010), the only parameter we adjusted was the number of split-merge iterations, which defaults to 6.

Petrov (2010) observed that the choice of random seed on this parser can yield substantial differences in the resulting model, and suggested that this is a consequence of the EM algorithm finding different local optima. Petrov proposes a procedure which creates a product model from eight differently initialised parser models, achieving an increase in precision of 1.8%. The variance observed by Petrov (2010) suggests that future work should investigate whether similar behaviour exists on English and Chinese CCGbank data, and whether the product parser approach would yield superior models on the two CCG corpora.

To understand whether the random seed also greatly influences models trained on Chinese CCGbank, some of our reported figures use a second model trained with a different random seed.

### 6.3.3 Z&C: Transition-based CCG parsing

Steedman (2000) suggested two classes of bottom-up parsing algorithms as being particularly good matches for CCG parsing: CKY-style chart parsing, and shift-reduce parsing. While chart parsers are well-represented in the CCG literature (Hockenmaier, 2003; Clark and Curran, 2007b), shift-reduce parsers are not commonly used, despite the fact that the seminal description of CCG by Ades and Steedman (1982) suggested the connection between application of combinatory rules and the reduction of partial constituents.

Zhang and Clark (2009) observed that while chart parsers characterised the state-of-the-art in English parsing, with transition-based parsers lagging behind in accuracy, the converse situation was true of Chinese parsing. Training a shift-reduce parser with actions scored by a discriminative model, Zhang and Clark (2009) achieved the best EVALB results in the literature for a parser trained on PCTB 5.0.

Shift-reduce parsers operate over a queue of input lexical items, a stack of partial constituents, and a set of actions which shift lexical items onto the stack or reduce partial con-
stituents on the stack. Zhang and Clark (2011) shows that shift-reduce parsers can easily be adapted to CCG with a system of four actions: shift, a unary reduce (which applies a CCG unary rule to the topmost partial constituent), a binary reduce (for CCG binary rules), and a finish action which allows the model to decide when to terminate parsing once all input lexical items have been shifted. The shift-reduce parser described by Zhang and Clark (2011) realised the first wide-coverage shift-reduce CCG parser in the literature, and showed that even with a narrow beam, the performance of a shift-reduce CCG parser could approach that of the best-performing C&C model.

Because the Z&C parser uses the C&C supertagger, the same features for supertagger training and operation given in Table 6.2 also apply.

Zhang and Clark (2011) operate the C&C supertagger in a high-ambiguity mode, with $\beta = 0.0001$. This is to increase the number of categories returned for each lexical item, since the shift-reduce parser is only able to compute analyses over the categories which the supertagger returns. During training, they also supplement the supertagger output with the gold-standard lexical category if the gold-standard category is not present in the supertagger output. By contrast, operating with the supertagger at $\beta = 0.0001$ caused training time to increase greatly and coverage to drop precipitously regardless of the value of $\tau$. As such, we used $\beta = 0.01$ for all our Z&C experiments.

Zhang and Clark also investigated changing the beam size, but found that while an increase in the beam size could increase performance at the cost of training and testing time, since the parser could explore more of the parsing space, performance did not increase dramatically beyond a beam size of 16. We use the default beam size, but consider that the optimal beam size for Chinese CCGbank may differ from that of English.

Z&C constrains the set of available reduce operations to those attested in the training data, also known in the C&C parser as the “seen rules” configuration. As such, Z&C is not capable of “full” CCG parsing, which can hypothesise arbitrarily many categories, including those never attested in its training data. As Fowler and Penn (2010) note, this is not typically a problem; many CCG parsers make this assumption to vastly increase the tractability of the parsing algorithm. However, unlike C&C, which automatically tabulates the rules attested in the training data as part of the training process, Z&C must be supplied a rules file which specifies the space of its reduce rules. Each production rule must additionally be annotated with its headedness; this affects whether the REDUCE RIGHT or REDUCE LEFT parser action is used. We are able to provide this headedness annotation, as the Chinese CCGbank conversion process uses and maintains headedness information at each internal
6.3. Approaches to ccg parsing

The Chinese CCGbank fork of the C&C distribution includes a script to generate the Z&C rules file.

This process introduces a threshold parameter $\tau$, which reflects the number of times a rule must be attested in the corpus before it is included in the rules file. This parameter controls a number of interesting trade-offs in the Z&C parser. The lower the value of $\tau$, the larger the rules file, and the less efficient training and parsing becomes, since the size of the rules file is the number of available reduce actions. Furthermore, a low threshold introduces noisier rules, but increases coverage by decreasing the number of sentences for which Z&C fails to find a spanning analysis due to rare but legitimate rules. The effect of $\tau$ on coverage can be seen in Figure 6.3, which plots $\tau$ against coverage on the left $y$-axis (the blue line with square points), and $F$-score on the right $y$-axis (the purple line with round points).

Figure 6.3 allows us to choose a value of $\tau$ which yields a high-coverage, high-accuracy model. We select a value $\tau = 39$ for testing, optimised on the development set.

---

2 The script which extracts a rule file is not included in the ZPAR distribution; users training a grammar should develop their own (Yue Zhang, p.c.).
6.4 Components common to all three parsers

In order for the output of the CCG parsers under test to be comparable, they must all accept the same training data and test data, and produce the same standard evaluation. To ensure this, Fowler and Penn (2010), converted the derivations produced by the P&K parser into CCG dependencies using the same tool which C&C uses to perform the same conversion. This tool, generate, is part of the C&C distribution, but as Fowler and Penn (2010) observe, is a deterministic conversion process which does not make use of any C&C parsing model, and is therefore suitable for use in an evaluation pipeline. This use of generate is similar to existing scripts which generate dependencies from gold derivations for training.

In turn, generate uses a file format known in C&C as markedup, which instantiates the head co-indexation mechanism introduced in Chapter 1. It specifies the head co-indexation behaviour of each category in the lexicon, and in the English grammar, also maps the dependencies arising from categories to the formalism-independent DepBank (King et al., 2005) scheme.

Finally, we present the dependency-based evaluation itself, which since its introduction in Clark and Hockenmaier (2002), has been the standard evaluation metric over CCG parsers. We describe the alternatives to this dependency-based evaluation metric, and describe how it can be extracted from candidate parses.

6.4.1 The markedup file

The coindexation within categories is defined in C&C by the markedup file. The markedup file annotates each structural category with its head coindexation, for example:

$$\left[(NP_{y} / NP_{y})_{x} \setminus (S[\text{dcl}]_{z} \setminus NP_{y})_{x}\right] \rightarrow ((NP_{y} / NP_{y})_{x} / (S[\text{dcl}]_{z} \setminus NP_{y})_{x})_{x} / (S[\text{dcl}]_{z}) / ((NP_{y})_{x} / (S[\text{dcl}]_{z} \setminus NP_{y})_{x}) / (NP_{y})_{x}$$

The C&C markedup file has the limitation that each supertag string maps to at most one markedup entry. Hockenmaier and Steedman (2005) notes that this limitation, which is also present in English CCGbank, prevents it from distinguishing subject- and object-control verbs, which should yield different markedup entries:

(1) a. promise $\vdash (S[\text{dcl}] / NP_{y}) / (S[to] \setminus NP) / NP_{y}$
   b. persuade $\vdash (S[\text{dcl}] / NP) / (S[to] \setminus NP_{y}) / NP_{y}$

However, because all instances of control identified in English CCGbank were object-control, the corpus only exhibits the second coindexation (i.e. persuade), which is the only entry in the C&C markedup file for the English grammar. As such, C&C trained on English
6.4. Components common to all three parsers

CCGbank returns incorrect dependencies in subject-control sentences. For instance, even when the parser is given gold-standard supertags, an incorrect dependency between Jade and feed would be returned in the following sentence:

(2) John promised Jade to feed Bec.

While the input tags for the C&C supertagger are conventionally a string representation of the category (that is, \( NP/\) → \( NP/NP \)), C&C does not require this. The markedup file simply maps the string representation for a supertag, whatever its textual content, to a structural category.

6.4.2 One structural category maps to multiple markedup entries

Similar to the subject-/object-control example above, the case also arises in the Chinese CCGbank grammar where one structural category must map to more than one markedup entry. This can be seen in the following example, in which 可能 might and 继续 continue share the structural category \( (S[dcl]/NP)/(S[dcl]/NP) \):

might be misused

\[
\begin{align*}
\text{可能} & \quad \text{被} \quad \text{误用} \\
(S[dcl]/NP)/(S[dcl]/NP) & \rightarrow (S[dcl]/NP)/(S[dcl]/NP) \sim SB \rightarrow S[dcl]/NP \\
\end{align*}
\]

continue to maintain its status as a financial centre

\[
\begin{align*}
\text{继续} & \quad \text{保持} \quad \text{国际} \quad \text{金融} \quad \text{中心} \quad \text{地位} \\
(S[dcl]/NP)/(S[dcl]/NP) & \rightarrow (S[dcl]/NP)/(S[dcl]/NP) \rightarrow S[dcl]/NP \\
\end{align*}
\]

Above, the non-gapped short 被 bei category coincides structurally with the usual category for two-place control verbs: \( (S[dcl]/NP)/(S[dcl]/NP) \). However, as with the subject- and
object-control categories in the English CCGbank grammar, the two categories are distinct with respect to their head variables:

(4) a. 被 bei ⊢ (S[decl]\NP_y)/(S[decl]\NP_z)
   b. 批准 allow ⊢ (S[decl]\NP_y)/(S[decl]\NP_z)

That is, the category of 被 bei does not induce a long-distance dependency between the subject of its VP complement and the subject of 被 bei, while the category of a two-place control verb does.

To allow a parser to distinguish these two cases, we post-process Chinese CCGbank according to Algorithm 16. We attach the annotation ~SB to the short 被 bei category so that it can be distinguished from the category of a two-place control verb in the supertagger training data. Note that this annotation is only visible to two components in the parser: the supertagger and the markedup file.

**Algorithm 16** Distinguishing two markedup entries using a category annotation

1. for every derivation D in the corpus do
2.   for every leaf L in derivation D do
3.     if the category of L is (S[decl]\NP_y)/(S[decl]\NP_z) and the pos tag of L is SB then
4.       rewrite the category of L to (S[decl]\NP_y)/(S[decl]\NP_z)~SB

### 6.4.3 Format of the markedup file

The markedup file consists of markedup entries, formatted as in Figure 6.4, separated by blank lines. Each entry begins with the supertag string to be mapped to that entry (for example, (NP/NP)/M). This is followed by a line which specifies the category annotated with head variables, preceded by the number of argument slots in that category. Each sub-part of the category must be followed by an annotation \{S\}, where S represents a head variable. In C&C, by convention the first head variable (denoted _) is bound to the lexical item of the category, and further head variables are used in the order YZWVUTRQABCDEF. Furthermore, the sub-parts of the category which correspond to argument slots are followed by the annotation <i>, where i is the index of the argument slot.\(^3\)

\(^3\) In the English markedup file, these lines followed by directives which convert the dependencies to the parser-independent DepBank King et al. (2003) format. We do not support this evaluation, and omit these lines.
6.4. Components common to all three parsers

\[(S[dcl]NP)/(S[dcl]NP)/NP\]

\[2 ((S[dcl]NP{Y}<1>)/((S[dcl]NP{W}){Z}/NP{Y}){Z}<2>)\]

Figure 6.4: Markedup entry for category \((S[dcl]NP)/(S[dcl]NP)/NP\)

6.4.4 Generating the markedup file automatically

Included with the C&C distribution is a markedup file for English CCGbank. While Clark and Curran (2007b) and Hoffman (1996) describe the manual annotation of coindexation for English and Turkish respectively, we present an algorithm to automatically generate much of the markedup file by examining category structure (Algorithm 17).

Line 1 of the algorithm asks if a cached markedup entry exists. This is to facilitate the annotation of cases which Algorithm 17 is not equipped to handle, such as categories which induce non-local dependencies. In general, markedup entries for such categories cannot be generated automatically: such an algorithm would somehow have to distinguish between the two coindexations in (4), based on the annotation in the source corpus. We manually create the markedup entries for categories which should induce non-local dependencies.

We also provide manual markedup entries for some categories, because Algorithm 17 yields incorrect results due to the heuristics it uses to distinguish modifier categories from argument-taking categories. An example of such a category is \((S[dcl]\ S[dcl])/NP\) — the category of a Chinese verb which takes an IP subject and an NP object argument. Without the manual markedup, Algorithm 17 would derive the markedup entry:

\[(5)\ [(S[dcl]_{y}S[dcl]_{y})/NP_{w}]\]

because \(S[dcl]\ S[dcl]\) has the form of a modifier category: \(X|X\) for some category \(X\). Instead, we force the correct markedup \((S[dcl]_{y}S[dcl]_{y})/NP\) by providing a manual coindexation.

Similarly, the question-forming particle \(\text{吗}\ ma\) in Chinese CCGbank is a functor from a declarative sentence to a yes-no question: \(S[q]\ S[dcl]\). For this category, Algorithm 17 yields the markedup \((S[q]_{y}S[dcl]_{y})\). Under this category, the head of a yes-no question marked by the particle \(\text{吗}\ ma\) is the particle itself. However, to force the coindexation \(S[q]_{y}S[dcl]_{y}\), so that the head of the outermost variable at the root of the derivation is
the main verb, the markup is specified manually. Table 6.3 shows the mapping from structural categories to markup categories for these special cases.

Table 6.3 shows the mapping from structural categories to markup categories for these special cases.

Algorithm 17 Markedup generation algorithm, mkmarked

Given: \( C \), a category; \( vars \), a sequence of free head variable labels

Note: \( vars \) is initially \( \langle Y, Z, W, \ldots, F \rangle \), the list of free head variable labels

1. if a cached markedup entry exists for \( C \) then
2. return a copy of it
3. else
4. if outermost head variable of \( C \) does not have a label then
5. \( C\text{-SLOT}.\text{VAR} \leftarrow \) next available variable label from \( vars \)
6. if \( C \) is a complex category then
7. \( cur \leftarrow C \)
8. while \( cur \) is a complex category and \( cur \) is neither a modifier category nor \( NP/N \) do
9. \( cur\text{-RES}.\text{SLOT} \leftarrow cat\text{-SLOT} \)
10. \( cur \leftarrow cur\text{-RES} \)
11. if \( C \) is a modifier category then
12. \( C\text{-RES} \leftarrow \text{LABEL}(C\text{-RES}, vars) \)
13. else
14. \( C\text{-RES} \leftarrow \text{LABEL}(C\text{-RES}, vars) \)
15. \( C\text{-ARG} \leftarrow \text{LABEL}(C\text{-ARG}, vars) \)
16. return \( C \)

6.4.5 Evaluating the automatic markedup file generation algorithm

We demonstrate that the manual effort undertaken by Clark and Curran (2007b) to create the markedup file can largely be obviated by the automatic markedup generation algorithm we presented as \texttt{mkmarked} (Algorithm 17).

We took the 580 categories in the C&C English markedup file and allowed \texttt{mkmarked} to generate the head co-indexation automatically. Just as Clark and Curran (2007b) manu-

\[ \frac{7}{5} \]
<table>
<thead>
<tr>
<th>Structural category</th>
<th>Markedup category</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N/N) \ (S[decl] \ NP)</td>
<td>([N/N] \ (S[decl] \ NP) \ (S[decl] \ NP))</td>
</tr>
<tr>
<td>(N/N) \ (S[decl] \ NP)</td>
<td>([N/N] \ (S[decl] \ NP) \ (S[decl] \ NP))</td>
</tr>
<tr>
<td>(NP/NP) \ (S[decl] \ NP)</td>
<td>([NP/NP] \ (S[decl] \ NP) \ (S[decl] \ NP))</td>
</tr>
<tr>
<td>(NP/NP) \ (S[decl] \ NP)</td>
<td>([NP/NP] \ (S[decl] \ NP) \ (S[decl] \ NP))</td>
</tr>
<tr>
<td>((NP/NP) / (NP/NP)) \ (S[decl] \ NP)</td>
<td>(((NP/NP) / (NP/NP)) \ (S[decl] \ NP) \ (S[decl] \ NP))</td>
</tr>
<tr>
<td>((NP/NP) / (NP/NP)) \ (S[decl] \ NP)</td>
<td>(((NP/NP) / (NP/NP)) \ (S[decl] \ NP) \ (S[decl] \ NP))</td>
</tr>
<tr>
<td>((NP/NP) / (NP/NP)) \ (S[decl] \ NP)</td>
<td>(((NP/NP) / (NP/NP)) \ (S[decl] \ NP) \ (S[decl] \ NP))</td>
</tr>
<tr>
<td>(S[decl] \ NP) \ (S[decl] \ NP)</td>
<td>([S[decl] \ NP] \ (S[decl] \ NP) \ (S[decl] \ NP))</td>
</tr>
<tr>
<td>(S[decl] \ NP) \ (S[decl] \ NP)</td>
<td>([S[decl] \ NP] \ (S[decl] \ NP) \ (S[decl] \ NP))</td>
</tr>
<tr>
<td>(S[decl] \ S[decl]) / S[decl]</td>
<td>([S[decl] \ S[decl]] \ (S[decl] \ S[decl]) \ (S[decl] \ S[decl]))</td>
</tr>
<tr>
<td>(S[decl] \ S[decl]) / S[decl]</td>
<td>([S[decl] \ S[decl]] \ (S[decl] \ S[decl]) \ (S[decl] \ S[decl]))</td>
</tr>
<tr>
<td>(S[decl] \ S[decl]) / PP</td>
<td>([S[decl] \ S[decl]] \ (S[decl] \ S[decl]) \ (S[decl] \ S[decl]))</td>
</tr>
<tr>
<td>(S[decl] \ S[decl]) / QP</td>
<td>([S[decl] \ S[decl]] \ (S[decl] \ S[decl]) \ (S[decl] \ S[decl]))</td>
</tr>
<tr>
<td>(S[decl] \ S[decl]) / M</td>
<td>([S[decl] \ S[decl]] \ (S[decl] \ S[decl]) \ (S[decl] \ S[decl]))</td>
</tr>
<tr>
<td>S[decl] \ S[decl]</td>
<td>(S[decl] \ S[decl]) \ (S[decl] \ S[decl])</td>
</tr>
<tr>
<td>(S \ S) \ (S \ S)</td>
<td>([S \ S] \ (S \ S) \ (S \ S))</td>
</tr>
<tr>
<td>(S \ S) \ (S \ S)</td>
<td>([S \ S] \ (S \ S) \ (S \ S))</td>
</tr>
<tr>
<td>(S \ S) \ (S \ S)</td>
<td>([S \ S] \ (S \ S) \ (S \ S))</td>
</tr>
<tr>
<td>(S \ LCP) \ (S \ NP)</td>
<td>([S \ LCP] \ (S \ NP) \ (S \ NP))</td>
</tr>
<tr>
<td>(S \ LCP) \ (S \ NP)</td>
<td>([S \ LCP] \ (S \ NP) \ (S \ NP))</td>
</tr>
<tr>
<td>(S \ NP) / (S \ NP)</td>
<td>([S \ NP] \ (S \ NP) \ (S \ NP))</td>
</tr>
<tr>
<td>(S \ NP) / (S \ NP)</td>
<td>([S \ NP] \ (S \ NP) \ (S \ NP))</td>
</tr>
<tr>
<td>(S \ NP) / (S \ NP)</td>
<td>([S \ NP] \ (S \ NP) \ (S \ NP))</td>
</tr>
<tr>
<td>(S \ NP) / (S \ NP)</td>
<td>([S \ NP] \ (S \ NP) \ (S \ NP))</td>
</tr>
<tr>
<td>(S \ NP) / (S \ NP)</td>
<td>([S \ NP] \ (S \ NP) \ (S \ NP))</td>
</tr>
<tr>
<td>(S \ NP) / (S \ NP)</td>
<td>([S \ NP] \ (S \ NP) \ (S \ NP))</td>
</tr>
<tr>
<td>(S \ NP) / (S \ NP)</td>
<td>([S \ NP] \ (S \ NP) \ (S \ NP))</td>
</tr>
<tr>
<td>(S \ NP) / (S \ NP)</td>
<td>([S \ NP] \ (S \ NP) \ (S \ NP))</td>
</tr>
<tr>
<td>(S \ NP) / (S \ NP)</td>
<td>([S \ NP] \ (S \ NP) \ (S \ NP))</td>
</tr>
<tr>
<td>(S \ NP) / (S \ NP)</td>
<td>([S \ NP] \ (S \ NP) \ (S \ NP))</td>
</tr>
<tr>
<td>(S \ NP) / (S \ NP)</td>
<td>([S \ NP] \ (S \ NP) \ (S \ NP))</td>
</tr>
<tr>
<td>(S \ NP) / (S \ NP)</td>
<td>([S \ NP] \ (S \ NP) \ (S \ NP))</td>
</tr>
</tbody>
</table>

Table 6.3: Markedup categories for NLDS and exceptional cases in Chinese CCGbank
ally annotated all the lexical categories occurring in English CCGbank at least 10 times, we generate the markedup for categories occurring in Chinese CCGbank at least 4 times. The change in threshold reflects the smaller size of the Penn Chinese Treebank, and was selected empirically to achieve coverage comparable to English C&C models.

We then compared the co-indexations output by the manual and automatic methods, judging \textit{markedup} output to have \textit{matched} the manual output if its co-indexation was unifiable with the manual coindexation.\footnote{We added special head coindexation to handle the special head-structure of the English determiner the $\dashv$ \textit{NP\{nb\}_y/N_y}. Without this special case, the algorithm would assign the coindexation \textit{NP\{nb\}_y/N_z}; since it does not account for the special relationship between \textit{NP} and \textit{N}, the algorithm would otherwise treat them as unrelated atoms.}

For instance, the pair of categories in the first row has unifiable coindexations, while the one in the second row does not:

\begin{align*}
\text{[(}NP_{y/N_y} \setminus NP_{z/N_z}\text{)}/(S[dcl]_w/NP_{z/N_z})_w]_y & \quad \text{[(}NP_{z/N_z} \setminus NP_{w/N_w}\text{)}/(S[dcl]_v/NP_{w/N_w})_v]\_} \\
\text{[(}NP_{y/N_y} \setminus NP_{z/N_z}\text{)}/(S[dcl]_w/NP_{z/N_z})_w]_y & \quad \text{[(}NP_{y/N_y} \setminus NP_{z/N_z}\text{)}/(S[dcl]_w/NP_{z/N_z})_w]_y}
\end{align*}

The co-indexations matched 437 (75\%) of the time. As suggested in Section 6.4.4, categories which enact special head structure or which mediate long-distance dependencies must have their co-indexation specified by the implementor. We exhaustively examined the 143 cases in which \textit{markedup} output did not match the C&C annotation, to determine whether these differences arose from errors in the algorithm, or involved categories whose co-indexation needed to be specified manually.

The causes for the 143 remaining cases differing from the output of the procedure \textit{markedup} in Algorithm 17 are given in Table 6.4. The most frequent case is that, as in the two coindexations in (1), a manual annotator must specify the coindexation which mediates a non-local dependency (116 cases). The next most frequent case involves structural categories $X|X$ which do not have the head coindexation categories — an example of this is the inner category in (5), $S[dcl] \setminus S[dcl]$. Six categories have the shape $(X|X)|X$, the result category of which \textit{markedup} identifies incorrectly as a modifier category; this error category is similar to the previous one. Three categories mediate bounded non-local dependencies, such as those arising from the English \textit{tough}-construction, and require manual annotation. Two categories: $NP/N[num]$ and $N/NP$, require the same exceptional coindexation as in the determiner category $NP\{nb\}_y/N_y$, in which the head of the non-bare \textit{NP} is the same as the head of the inner bare \textit{NP}. Finally, one category — $(PP/PP)/(PP/PP)\setminus (S[adj] \setminus NP)$ — has a malformed entry in the markedup file.
6.4. Components common to all three parsers

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>116</td>
<td>Category mediates NLD, requiring manual annotation</td>
</tr>
<tr>
<td>15</td>
<td>Non-modifier categories with modifier shape</td>
</tr>
<tr>
<td>6</td>
<td>Categories with ((X</td>
</tr>
<tr>
<td>3</td>
<td>Category mediates bounded non-local dependency</td>
</tr>
<tr>
<td>2</td>
<td>NP or N special coindexation</td>
</tr>
<tr>
<td>1</td>
<td>Malformed in C&amp;C markedup</td>
</tr>
</tbody>
</table>

Table 6.4: Breakdown of 145 cases in which automatic markedup annotations differ

The manual error analysis confirms that in every case for which \texttt{mkmarked} did not match the manually derived C&C markedup annotation, the algorithm obtained the correct head coindexation. This shows that an automatic algorithm was automatically able to recover the head coindexation of a full 75\% of English CCGbank categories in a way which matches the manual annotation. Secondly, as a side benefit, we have validated the manually annotated C&C markedup file, even finding a malformed head coindexation.

6.4.6 generate tool

The \texttt{generate} tool is included in C&C to produce labelled word-word dependencies, or DepBank output if the markedup file contains the annotations to generate it, from derivations in the CCGbank format. In C&C, \texttt{generate} is used during training to extract gold-standard dependencies from the training data. As Fowler and Penn (2010) note, since \texttt{generate} does not use a C&C parser model in any way, it can be integrated into other parsers, for instance to convert the output of other parsers to the uniform format expected by the \texttt{cctg} evaluation over labelled dependencies used by the C&C parser.

\texttt{generate} operates by reading in the head coindexations specified in the markedup file, then performing the unification operations which arise from the action of combinatory rules in order to generate predicate-argument dependencies from a \texttt{cctg} derivation. It is important to note that \texttt{generate} can fail for several reasons: \textit{invalid unary/binary rule}, which occurs when the derivation input to \texttt{generate} contains a unary or binary rule which is not implemented in the C&C ruleset; or \textit{missing markedup}, which occurs when a derivation contains a category missing from the markedup file. In neither case can \texttt{generate} proceed; both cases prevent the process from passing head variables. Therefore, sentences can incur coverage loss because of issues arising from \texttt{generate}. When we evaluate coverage loss
later in the chapter, we distinguish parser coverage losses (where the parser has failed to produce any analysis) from generate coverage losses (where generate has failed for one of the above reasons).

We follow Fowler and Penn (2010) in using generate to extract the CCG evaluation format from the output of the P&K and Z&C parsers.

### 6.4.7 CCG dependency-based evaluation

Parser evaluation schemes are a basis by which the quality of the output of parsers, regardless of their architecture or base formalism, can be compared. As a proxy for native speaker judgement, in all practical parser evaluations, the candidate analyses output by parsers are compared against gold standard analyses. The free variables of this methodology include how to compute the similarity between a candidate analysis and the gold standard analysis; whether to transform the candidate analysis, the gold standard analysis, or both, and into what form; what kind of gold standard corpus to use; and how to aggregate performance over individual sentences into a summary statistic.

The PARSEVAL metric (Harrison et al., 1991; Black et al., 1991, 1992), also known as evalb, arose from the observation that output for a number of Brown corpus sentences, manually bracketed by a group of linguistically-aware annotators, very rarely matched exactly. They concluded that many attachment decisions, such as the attachment of punctuation, negation words and adverbs, are artifacts of the base formalism, and a useful evaluation metric should not penalise a candidate parse if it diverges from the gold standard parse in respects which reflect systematic formalism-internal differences. The resulting metric considers only recall of bracketings relative to the gold standard, and whether the candidate bracketings cross those of the gold standard tree. This approach penalises divergent bracketing choices, at the cost of levelling distinctions in category labels, and any encoding of long-distance dependencies. Furthermore, the normalisation steps which try to correct for formalism-internal differences are language-dependent. Alternatives to evaluating on PARSEVAL include cross-formalism dependency schemes such as Stanford dependencies (de Marneffe et al., 2006) and DepBank (King et al., 2003), as well as categorising the resulting errors by type Kummerfeld et al. (2012).

Despite, or because of, the compromises PARSEVAL makes to ease comparability, it has become the de facto standard evaluation for constituent parsers.

Manning and Carpenter (2000) observed that any metric which only penalises crossing bracketings ignores the difficult problem of bracketing NP internal structure when, like the
Penn Treebank, such structure is underspecified in the corpus. They also point out that given the Penn Treebank bracketing scheme, PARSEVAL fails to discriminate between the alternatives in the English PP attachment ambiguity.

Carroll et al. (1998) argues against PARSEVAL in favour of a dependency-based evaluation, proposing a scheme of cross-linguistic grammatical relations (GRS) as the medium of comparison. The proposed summary metric is $F$-score computed over candidate and gold GR tuples. The scheme generalises over formalism-internal differences in representation, and is sensitive to a parser’s ability to capture long-distance dependencies. One cost of such a scheme is that evaluating dependency parsers involves mapping from an internal dependency format to a particular GR scheme, and constituent parsers must convert their bracketed output to compatible dependency output using a scheme such as Johansson and Nugues (2007).

To evaluate their C&C parser, Clark and Curran (2007b) consider a native CCG-specific dependency-based metric which is then mapped onto Carroll et al.-style dependencies for cross-parser comparison. This metric, which is standard in the literature for comparing CCG parsers, like the GR scheme of Carroll et al., computes $F$-score over dependency tuples. However, in Clark and Curran (2007b) the dependency arcs are labelled not with cross-linguistic grammatical relations, but with a CCG category and slot. This metric is particularly unforgiving, since it punishes any dependency with a head whose supertag is incorrect. For example, given the Chinese noun/verb ambiguity, wrongly determining the head’s noun/verb status will punish any attachments with that head.

A dependency-based evaluation metric fixes several parameters of the evaluation scheme, but leaves open the question: what kind of corpus should the evaluation range over? A common choice is to compute a summary statistic over a pre-agreed test corpus, such as Section 23 of the Penn Treebank. Rimell et al. (2009) argues that an aggregate summary statistic obscures a parser’s capability to retrieve classes of dependencies which are relatively uncommon but difficult to decide. They focus on several types of long-distance dependencies, creating an English evaluation corpus annotating specifically for these difficult dependency types. However, due to differences in the output of the parsers under comparison, they end up manually converting each parser’s output into their scheme, making an evaluation under their scheme accurate (since no conversion errors were incurred), but costly to perform.
6.5 Chinese POS tagging

The confluence of low morphology and categorial fluidity gives rise to confusion types in Chinese POS tagging which are not common in English. For example, the noun/verb ambiguity uncovered by Levy and Manning (2003) results from the absence in Chinese of morphological cues, which POS taggers for morphologically richer languages can use to discriminate one word class from the other given only local context.

While English POS tagging is widely considered solved in NLP, with the state-of-the-art achieving 97.50% on PTB sections 22–24 (Søgaard, 2011), the corresponding figure for Chinese is only 94.50% (Hatori et al., 2011). Li et al. (2011) suggest that the increased degree of ambiguity in Chinese may exacerbate the propagation of error through a pipeline tagger-parser, observing a gap of ~6% between the performance of a parser given gold POS tags and one performing automatic POS tagging. Later in this chapter, we observe a similar gap between the C&C parser’s auto and gold configurations, demonstrating that the availability of the ground truth annotation during parsing substantially increases parsing performance.

The three parsers we focus on in this work handle POS tagging in distinct ways. C&C consists of a pipeline of POS tagging, supertagging and parsing, with the latter two steps interacting through adaptive supertagging. One consequence of a pipeline system is that incorrectly judging ambiguities such as the kind arising from Chinese noun/verb fluidity introduces sources of error which are nearly impossible to recover from.

Describing the operation of the Z&C system, Zhang and Clark (2011) only implement a parser; the C&C POS tagger and supertagger are used to multi-tag the input data. Thus, Z&C is vulnerable to the same mode of error propagation as in C&C, since the tagger is entirely detached from the parser.

The P&K parser treats POS tags as the pre-terminal projections of lexical items during training and testing, and therefore performs joint tagging and parsing; Fowler and Penn (2010), who trained the P&K parser directly on English CCGbank data, does not use POS tags at all, because they train the parser on English CCGbank trees whose labels are all CCG category strings.
To evaluate whether improvements to the POS tagger yield significant performance improvements, we apply a non-parametric randomised paired significance test from Cohen (1995), written by Dan Bikel, who used this significance test in his EVALB comparator.7

The test is suitable for paired data, such as the output of two tagger models run on the same dataset, and determines the probability of incorrectly rejecting the null hypothesis that both sequences of paired output arise from the same tagger model. The test estimates this probability with the randomised procedure in Algorithm 18. Initially, the test computes the difference according to an evaluation metric between the two sets of results, relative to a gold standard. Then, in each iteration, the algorithm randomly permutes each pair of corresponding results with probability 0.5. Once each pair has been permuted (or not), the evaluation metric is re-computed, and the new difference between the two results compared to the original difference. A counter k incremented if the new difference exceeds or matches the original difference. Finally, the p-value is computed as in Algorithm 18. We follow Bikel’s implementation by using the default of 10 000 iterations.

### Algorithm 18 The stratified shuffling test

**Precondition:** a gold standard label sequence G, two candidate label sequences C1 and C2, an evaluation metric f(·, ·), and a number of iterations N

1. d ← f(G, C1) − f(G, C2)
2. k ← 0
3. for i ∈ {1, 2, . . . , N} do
   4. C1 ← copy of C1
   5. C2 ← copy of C2
   6. for j ∈ {1, 2, . . . , |C1|} do
      7. if rand > 0.5 then
         8. exchange the labels C1j and C2j
      9. d′ ← f(G, C1′) − f(G, C2′)
   10. if d′ ≥ d then
      11. k ← k + 1
5. return (k + 1)/(N + 1)

---

6.5.2 Prosodic features for POS tagging

We experiment with modifying the C&C POS tagger to capture prosodic constraints on Chinese word formation, an effect which limits the viability of compounding strategies.

Prosodic effects on compound formation have been observed in Chinese, in which the well-formedness of noun-noun, and verb-noun compounds is conditioned on the length in syllables of the compounded elements (Duanmu, 2000; Feng, 1998).

We present examples of prosodic constraints. The four examples below are all N + N compounds between morphemes meaning coal and shop. However, the prosody constraint operating on the following compounds decides whether each compound is well-formed:

(6) a. 煤炭 商店
meitan shangdian

b. 煤炭 店
meitan dian

c. 煤 店
mei dian

d. *煤 商店
mei shangdian
coal shop

If we write the combination of a one syllable morpheme and a two syllable morpheme as $1 + 2$, then the prosody constraint on $N + N$ compounds allows $1 + 1, 2 + 1, 2 + 2$, but blocks $1 + 2$. A different prosodic effect can also be observed in $V + O$ (verb plus object) compounds:

(7) a. 购买 车票
goumai chepiao

b. 买 车票
mai chepiao

c. 买 票
mai piao

d. *购买 票
goumai piao
buy (train) tickets

---

8 With the notable exception of the rhotacising suffix -er, and novelties such as 図 tushuguan, Chinese hanzi map exactly one glyph to one syllable. Therefore, the length in syllables of a morpheme is, with these exceptions, its length in characters.
While the $1 + 2$ combination is blocked for $N + N$ compounds, it is the $2 + 1$ combination which is blocked in $V + O$ compounds.

Thus, prosodic constraints can resolve the $V + O$ and $N + N$ ambiguity:

\[(8) \quad \text{Jilin Changchun open for development economic applicable house}\]

Changchun, Jilin opens up development of affordable housing

While 适用 applicable has a stative verb sense (be applicable), the verbal interpretation should be blocked by prosodic constraints, since its putative object 房 house is monosyllabic. We investigate whether supplying prosodic information to a tagger could aid in discriminating $V + O$ and $N + N$ compounds, which are a notable error case in Chinese caused by its low morphology.

We experimented with the following encoding of prosody. The binned length of each lexical item $w$ is $BL(w) = \min\{3, |w|\}$, where $|w|$ is the length in characters of $w$. This is because the literature on prosodic constraints does not appear to ascribe effects to morphemes which are not either monosyllabic or disyllabic. We add the following history feature to the POS tagger, where $w_0$ is the current word, and $w_{-1}$ is its predecessor:

\[
\langle \text{POS tag, } BL(w_{-1}), BL(w_0) \rangle
\]

We also added a feature which detects whether a lexical item has one of the reduplication patterns AABB, ABAB, AAB, ABB, or AA. The reduplication patterns AABB and ABAB are known to distinguish verbs from adjectives in Chinese, since the reduplication strategy ABAB is not available to adjectives, and the strategy AABB is not available to verbs (Huang et al., 2008):

\[(9)\]

a. *检检查查
   \textit{jianjian-chacha}
   have a check

b. 检检查查
   \textit{jiancha-jiancha}

\[(10)\]

a. 干干净净
   \textit{gangan-jingjing}
   clean and tidy

b. *干净干净
   \textit{ganjing-ganjing}

We then evaluated the tagger on our split of the PCTB 6.0 development set, with all combinations of the reduplication feature and prosodic feature, then tested each of the three
Chapter 6. Parsing Chinese CCGbank

<table>
<thead>
<tr>
<th>Prosodic feature</th>
<th>Redup feature</th>
<th>Accuracy</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>92.68%</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>✓</td>
<td>92.69%</td>
<td>0.50</td>
</tr>
<tr>
<td>✓</td>
<td>-</td>
<td>92.75%</td>
<td>0.14</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>92.71%</td>
<td>0.34</td>
</tr>
</tbody>
</table>

Table 6.5: Results of POS tagger trained with prosodic and reduplicative features

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>All distance features in Table 6.2</td>
<td>89.10%</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>Base model (bitags only)</td>
<td>87.95%</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6.6: The effect of extended horizon features on supertagging: two models evaluated on the PCTB 6.0 development set

configurations against the base configuration with Bikel’s stratified shuffling test, yielding the results in Table 6.5. The differences in accuracy between any of the three models and the base model were not statistically significant according to the stratified shuffling test.

6.6 Chinese supertagging

The average length of a PCTB sentence is about 15% longer than that of a PTB sentence. Accordingly, we decided to experiment with increasing the size of the C&C supertagger window, anticipating that a larger supertagger window may be needed to prevent important context from being pushed out of the tagger’s horizon.

We experimented with adjusting the size of the C&C supertagger’s context window by adding extended horizon features. These features were first implemented by Kummerfeld (2009) for English CCG parsing, increasing the size of the context window to seven tokens (three on either side) from five. Kummerfeld found that having trained the supertagger on four million words from the North American News Corpus, the far horizon features had little effect, or in the case of the sparser triwords/tritags features, degraded performance.

The C&C flags we use to control the horizon features are included in Table 6.2. To quantify the effect of additional context on supertagging models, we trained two models on the PCTB 6.0 training set on Chinese CCGbank, evaluating on the development set. One
model had all additional distance features active (all of the features in the “Supertagger window features” section of Table 6.2), while the other only had the bitags feature active (the default configuration). As in the pos tagging experiment, statistical significance is determined by Bikel’s compute-intensive stratified shuffling tests. The results of the two models, including the significance of the large-horizon model compared to the base model, are given in Table 6.6. The difference between the two models is highly statistically significant according to the stratified shuffling test.

While Kummerfeld (2009) found that the extended horizon features had little impact on English supertagging performance, we observe a substantial improvement for Chinese, reinforcing the need for further analysis to understand how the characteristics of Chinese influence the effectiveness of nlp techniques.

### 6.7 Models for Chinese parsing

The C&C parser estimates a log-linear model, given features which capture properties of the candidate derivation. The maximum entropy framework flexibly incorporates overlapping features, which we exploit to improve the Chinese parsing model.

We reproduce the list of core C&C parsing features from Clark and Curran (2007b) in Table 6.9, built from the feature functions in Table 6.7. Each row is a feature value, which may comprise multiple values. For example, the feature \( \langle L(h), P(a) \rangle \) is the concat-
enation of the lexical item of a head with the pos tag of its argument, while the feature \( C(l) C(r) \rightarrow C(p) \) consists of the categories of the left child, right child and parent concatenated together. The feature index is the internal C&C representation of the feature, which we give in Table 6.7 for the benefit of implementors working with C&C.

For our parser experiments, we implemented three new features, given at the bottom of the first column in Table 6.7. The first two (feature indices 3 and 4) concatenate the pos tag or lexical item of the head with a one-character suffix of the argument. This exploits the largely right-headed word compounding strategy of Chinese (Packard, 2000), through the observation that a one-character suffix in Chinese is often the morphological head of the lexical item:

\[
\begin{align*}
\text{a.} & \quad \text{自 } - \text{行-} - \text{车} \\
& \quad \text{self} - \text{powered} - \text{car} \\
& \quad \text{bicycle}
\end{align*}
\]

\[
\begin{align*}
\text{b.} & \quad \text{崇明-} - \text{县} \\
& \quad \text{Chongming} - \text{province} \\
& \quad \text{Chongming Province}
\end{align*}
\]

We also added a feature (index j) which concatenates the head lexical item with the length in characters of the argument’s lexical item, clamped to three. This captures the prosodic constraint on V-O compounds described in Section 6.5.2, in which monosyllabic verbs can take mono- or disyllabic objects, but disyllabic verbs can only take monosyllabic objects.

Table 6.8 gives the results of C&C parser models with these features added. We observe a small decrease in performance in labelled F-score from 67.37% to 67.19% when the suffix features are disabled, suggesting that these features allow some generalisation over lexical items. Subtracting the prosodic feature does not greatly influence the resulting model.

These very preliminary experiments suggest that some gains may result from tailoring parser features to the characteristics of Chinese text. We are aware of work by Xue and Yang (2011) which focuses on the different distribution of commas in Chinese, and Sun and Jurafsky (2003) who investigated an encoding of prosody for Chinese NP chunking, but to our knowledge work exploiting the properties of Chinese text is still under-explored.
### 6.7. Models for Chinese parsing

#### Table 6.8: C&C evaluation of extra dependency features on Chinese CCGbank dev set, AUTO POS tags

<table>
<thead>
<tr>
<th>Index</th>
<th>Feature type</th>
<th>Index</th>
<th>Feature type</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>(\langle C(w), L(w) \rangle)</td>
<td>m</td>
<td>(C(l) C(r) \rightarrow C(p))</td>
</tr>
<tr>
<td>b</td>
<td>(\langle C(w), P(w) \rangle)</td>
<td>n</td>
<td>(C(l) \rightarrow C(p))</td>
</tr>
<tr>
<td>c</td>
<td>(C(p))</td>
<td>p</td>
<td>(\langle C(l) C(r) \rightarrow C(p), L(h) \rangle)</td>
</tr>
<tr>
<td>d</td>
<td>(\langle P(p), H(p) \rangle)</td>
<td>q</td>
<td>(\langle C(l) \rightarrow C(p), L(l) \rangle)</td>
</tr>
<tr>
<td>e</td>
<td>(\langle L(h), L(a) \rangle)</td>
<td>r</td>
<td>(\langle C(l) C(r) \rightarrow C(p), P(h) \rangle)</td>
</tr>
<tr>
<td>f</td>
<td>(\langle P(h), P(a) \rangle)</td>
<td>s</td>
<td>(\langle C(l) \rightarrow C(p), P(l) \rangle)</td>
</tr>
<tr>
<td>h</td>
<td>(\langle L(h), P(a) \rangle)</td>
<td>t</td>
<td>(\langle L(l), C(l) C(r) \rightarrow C(p), L(r) \rangle)</td>
</tr>
<tr>
<td>i</td>
<td>(\langle P(h), L(a) \rangle)</td>
<td>u</td>
<td>(\langle P(l), C(l) C(r) \rightarrow C(p), L(r) \rangle)</td>
</tr>
<tr>
<td>L</td>
<td>(\langle L(h), P(a), D_w(h, a) \rangle)</td>
<td>v</td>
<td>(\langle L(l), C(l) C(r) \rightarrow C(p), P(r) \rangle)</td>
</tr>
<tr>
<td>P</td>
<td>(\langle P(h), P(a), D_w(h, a) \rangle)</td>
<td>w</td>
<td>(\langle P(l), C(l) C(r) \rightarrow C(p), P(r) \rangle)</td>
</tr>
<tr>
<td>M</td>
<td>(\langle L(h), P(a), D_v(h, a) \rangle)</td>
<td>F</td>
<td>(\langle L(l), C(l) C(r) \rightarrow C(p), P(r), D_w(l, r) \rangle)</td>
</tr>
<tr>
<td>Q</td>
<td>(\langle P(h), P(a), D_v(h, a) \rangle)</td>
<td>I</td>
<td>(\langle P(l), C(l) C(r) \rightarrow C(p), P(r), D_w(l, r) \rangle)</td>
</tr>
<tr>
<td>N</td>
<td>(\langle L(h), P(a), D_p(h, a) \rangle)</td>
<td>G</td>
<td>(\langle L(l), C(l) C(r) \rightarrow C(p), P(r), D_v(l, r) \rangle)</td>
</tr>
<tr>
<td>R</td>
<td>(\langle P(h), P(a), D_p(h, a) \rangle)</td>
<td>J</td>
<td>(\langle P(l), C(l) C(r) \rightarrow C(p), P(r), D_v(l, r) \rangle)</td>
</tr>
<tr>
<td>New Chinese features</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>(\langle P(h), S_1(a) \rangle)</td>
<td>H</td>
<td>(\langle L(l), C(l) C(r) \rightarrow C(p), P(r), D_p(l, r) \rangle)</td>
</tr>
<tr>
<td>4</td>
<td>(\langle L(h), S_1(a) \rangle)</td>
<td>K</td>
<td>(\langle P(l), C(l) C(r) \rightarrow C(p), P(r), D_p(l, r) \rangle)</td>
</tr>
<tr>
<td>j</td>
<td>(\langle L(h), B(a) \rangle)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.9: C&C parser features
6.8 Modifying the C&C parser for Chinese parsing

This section describes all of the modifications we made to the C&C parser to support Chinese CCG parsing. C&C consists of over 50,000 lines of highly optimised C++ code, and is tuned for performance rather than ease of modification.

Changes to categories. The C&C parser is optimised for English CCGbank, and categories have a representation optimised for space. As such, C&C contains enumerations representing atom types and feature types. A consequence of this compact representation is that changes to the atom or feature set require modifications of the category or feature representation. We removed all English categories and features, re-adding the atom and feature set we developed for Chinese in Chapter 2.

Changes to parser rules. Similarly, all of the binary rules are implemented in the parser in a large switch statement, which decides for a given pair of categories what combinations are possible. In order to add a new rule, the implementor must decide whether the rule should be active only during generate, in which the parser tries to recreate the gold standard parse in order to extract features. The reason that certain parser rules are only active during generate is to increase the coverage over the training data. These low-frequency rules are then turned off during parsing to increase efficiency by blocking rare analyses.

We removed all English CCGbank type-change rules, and added the rules listed in Appendix B. In C&C, type-raising rules are specified separately to unary rules. We add the type-raising rules licensed by the Chinese CCGbank grammar.

Representation of lexical items. As the Penn Chinese Treebank uses the multi-byte UTF-8 encoding, tagger or parser features which index into the byte representation of lexical items will yield incorrect results. We change the processing of lexical items to correctly parse the UTF-8 representation used by the PCTB into wide characters, allowing features to address logical characters instead of bytes.

Parser features. The internal representation of parser features is highly compact, because of the massive number of features typically extracted. In this compact representation, features are hashed for fast access, and are of fixed type. For instance, specialised feature classes are used to hold each of the C&C parser feature types in Table 6.9. To support ease of experimentation, we implemented a new feature class which allows arbitrary strings to be used as features.

---

9 The initial revision of the Chinese fork of C&C was revision 1424 of the C&C trunk repository.
6.9 Notes on the experimental setup

Experiments were done on a cluster, consisting of four 64-bit four-core Xeon E5520 processors at 2.27GHz and 16GB RAM each, and four 64-bit eight-core Xeon E5-2470 processors at 2.30GHz with 64GB RAM each. As described in Clark and Curran (2007b), training of the C&C parser is performed on a cluster using MPI (Message Passing Interface). We use the MPI version of supertagger training developed by Kummerfeld et al. (2009).

Training of both Z&C and P&K was performed single-core. While P&K has a flag -nThreads for parsing in parallel, we observed stability issues with the flag set, and so parsing was performed single-core as well. Wall-clock time estimates for training and parsing each parser are given in Table 6.10. The wall-clock time for training and testing Z&C includes time elapsed invoking the C&C multitagger.

Table 6.10 shows that training and testing is fastest on Z&C, owing to the efficiency of perceptron training and beam search. C&C tagger and parser training is intensive, even on a large cluster of 36 nodes, but tagging and parsing is relatively fast. P&K training is very slow in the sixth and last split-merge iteration, owing to the large number of split categories, and parsing takes a considerable amount of time as well.

\[ \tau = 39 \]

\[ \begin{array}{cccc}
\text{Component} & \text{Training time} & \text{Testing time} & \# \text{train nodes} \\
\hline
\text{C&C} & & & \\
\text{POS tagger} & 27 \text{ mins} & 1.8 \text{ secs} & 1 \\
\text{Supertagger} & 55 \text{ mins} & 5.5 \text{ secs} & 36 \\
\text{Parser} & 130 \text{ mins} & 9.8 \text{ mins} & 36 \\
\hline
\text{P&K} & & & \\
\text{I-6} & 714 \text{ mins} & 320 \text{ mins} & 1 \\
\hline
\text{Z&C} & & & \\
\tau & 14 \text{ mins} & 1.5 \text{ mins} & 1 \\
\end{array} \]

Table 6.10: Wall-clock time estimates for training and parsing

\[ ^{10} \text{These are point estimates of training and parsing time, and are intended only to give order-of-magnitude impressions of the processing time for each parser.} \]
Chapter 6. Parsing Chinese CCGbank

We are finally ready to present the standard evaluation for C&C, P&K and Z&C trained on Chinese CCGbank. Having presented the results, we dedicate the remainder of the text to understanding the causes of the observed differences between the three parsers.

Table 6.11 presents the results obtained from the three parsers trained on Chinese CCGbank. We also perform the same evaluation over the Chinese CCGbank test set in Table 6.12. Table 6.11 gives the standard evaluation over a limited test set consisting of all sentences parsed successfully by each pair of parsers (the intersection set evaluation).

For each parser under test, we give its LF (F-score over labelled dependencies), UF (F-score over unlabelled dependencies), Lsa% (the proportion of sentences for which the parser returned all and only the correct labelled dependencies), stag (supertagger accuracy) and cov (coverage). Several configurations of each parser are also used:

**C&C.** We provide gold (gold-standard pos tags) and auto (automatic pos tags).

**P&K.** Following Fowler and Penn (2010), we denote a model trained on k split-merge iterations as I-k. We also train models with the random seed 24601, to observe the impact of initialisation on the resulting model; this configuration is referred to as I-k*.

**Z&C.** We vary the rule threshold $\tau$.

Following Zhang and Clark (2011) and McDonald and Nivre (2007), we break down the accuracy of dependencies per parser by head-argument distance, allowing the aggregate behaviour of each parser on easier, local (shorter) dependencies to be compared to the behaviour on harder, non-local (longer) dependencies. This analysis is given in Figure 6.5.

<table>
<thead>
<tr>
<th>Parser</th>
<th>LF</th>
<th>UF</th>
<th>Lsa %</th>
<th>stag</th>
<th>cov</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z&amp;C ∩ C&amp;C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>au</td>
<td>67.62</td>
<td>78.73</td>
<td>13.99</td>
<td>83.89</td>
<td>100.0</td>
</tr>
<tr>
<td>$\tau = 39$</td>
<td>72.86</td>
<td>81.57</td>
<td>17.17</td>
<td></td>
<td>100.0</td>
</tr>
<tr>
<td>P&amp;K ∩ C&amp;C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>au</td>
<td>68.62</td>
<td>79.45</td>
<td>15.35</td>
<td>84.61</td>
<td>100.0</td>
</tr>
<tr>
<td>I-6</td>
<td>72.42</td>
<td>82.95</td>
<td>16.86</td>
<td>85.45</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Table 6.11: Evaluation only on sentences parsed by C&C and each other parser

6.10 Multi-parser empirical evaluation of Chinese CCGbank
### Table 6.12: Empirical evaluation of Chinese CCGbank on three parsers (dev set)

<table>
<thead>
<tr>
<th>Parser</th>
<th>LF</th>
<th>UF</th>
<th>Lsa %</th>
<th>stag</th>
<th>cov</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>C&amp;C</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GOLD</td>
<td>76.08</td>
<td>84.63</td>
<td>19.35</td>
<td>89.41</td>
<td>99.47</td>
</tr>
<tr>
<td>AUTO</td>
<td>67.37</td>
<td>78.55</td>
<td>13.98</td>
<td>84.00</td>
<td>98.84</td>
</tr>
<tr>
<td><strong>P&amp;K</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I-1</td>
<td>65.95</td>
<td>77.32</td>
<td>10.97</td>
<td>82.68</td>
<td>94.52</td>
</tr>
<tr>
<td>I-3</td>
<td>69.87</td>
<td>80.72</td>
<td>14.39</td>
<td>84.08</td>
<td>94.06</td>
</tr>
<tr>
<td>I-5</td>
<td>71.89</td>
<td>82.56</td>
<td>17.05</td>
<td>85.14</td>
<td>93.76</td>
</tr>
<tr>
<td>I-6</td>
<td>72.20</td>
<td>82.65</td>
<td>16.68</td>
<td>85.16</td>
<td>94.64</td>
</tr>
<tr>
<td>I-6*</td>
<td>71.88</td>
<td>82.63</td>
<td>15.55</td>
<td>84.92</td>
<td>93.37</td>
</tr>
<tr>
<td><strong>Z&amp;C</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\tau = 0$</td>
<td>74.86</td>
<td>83.38</td>
<td>19.86</td>
<td>–</td>
<td>73.22</td>
</tr>
<tr>
<td>$\tau = 2$</td>
<td>74.62</td>
<td>83.20</td>
<td>20.22</td>
<td>–</td>
<td>79.74</td>
</tr>
<tr>
<td>$\tau = 10$</td>
<td>72.55</td>
<td>81.33</td>
<td>17.49</td>
<td>–</td>
<td>92.69</td>
</tr>
<tr>
<td>$\tau = 20$</td>
<td>72.48</td>
<td>81.51</td>
<td>17.61</td>
<td>–</td>
<td>96.04</td>
</tr>
<tr>
<td>$\tau = 39$</td>
<td>72.55</td>
<td>81.27</td>
<td>17.04</td>
<td>–</td>
<td>98.24</td>
</tr>
</tbody>
</table>

### Table 6.13: Empirical evaluation of Chinese CCGbank on three parsers (test set)

<table>
<thead>
<tr>
<th>Parser</th>
<th>LF</th>
<th>UF</th>
<th>Lsa %</th>
<th>stag</th>
<th>cov</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>C&amp;C</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GOLD</td>
<td>76.63</td>
<td>84.80</td>
<td>22.70</td>
<td>89.54</td>
<td>99.00</td>
</tr>
<tr>
<td>AUTO</td>
<td>66.91</td>
<td>78.18</td>
<td>14.87</td>
<td>83.92</td>
<td>98.79</td>
</tr>
<tr>
<td><strong>P&amp;K</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I-6</td>
<td>71.94</td>
<td>82.64</td>
<td>17.39</td>
<td>85.07</td>
<td>94.47</td>
</tr>
<tr>
<td><strong>Z&amp;C</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\tau = 39$</td>
<td>73.26</td>
<td>81.81</td>
<td>19.60</td>
<td>–</td>
<td>98.02</td>
</tr>
</tbody>
</table>
Fig. 6.5: Performance by dependency length

(a) Labelled F-score by dependency length on Chinese CCGbank dev set

(b) Number of proposed dependencies by dependency length
6.10. Multi-parser empirical evaluation of Chinese CCGbank

(a) Recall by dependency length

(b) Precision by dependency length

Figure 6.6: $P$ and $R$ by dependency length
Table 6.14: Summary of Chinese parsing approaches

<table>
<thead>
<tr>
<th></th>
<th>English</th>
<th>Chinese</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTB/PCTB-based</td>
<td>92.1% (McClosky et al., 2006)</td>
<td>86.8% (Zhang and Clark, 2009)</td>
</tr>
<tr>
<td></td>
<td>85.5% (Zhang and Clark, 2011)</td>
<td>72.6% (Z&amp;C; Table 6.12)</td>
</tr>
<tr>
<td>CCGbank-based</td>
<td>86.0% (Fowler and Penn, 2010)</td>
<td>72.2% (P&amp;K; Table 6.12)</td>
</tr>
<tr>
<td></td>
<td>85.8% (Clark and Curran, 2007b)</td>
<td>67.4% (C&amp;C; Table 6.12)</td>
</tr>
</tbody>
</table>

6.11 Analysis of parsing results

It is informative to view the Chinese CCGbank parsing results of the previous section in the context of similar (but not directly comparable) results. Table 6.14 summarises the state-of-the-art in English and Chinese parsers trained on phrase structure (Penn Treebank) data or on CCGbanks. Not all figures from this table are directly comparable; we present them in a table to give an impression of the differences between similar results.

Firstly, differences between the states-of-the-art in Chinese and English treebank parsing (the pair of figures in the first row of Table 6.14) has long been observed, beginning with Bikel and Chiang (2000) and Levy and Manning (2003). This suggests that we should expect to see a roughly comparable deficit between English and Chinese CCG parsing as well.

An invaluable aspect of our multi-parser comparison is that each of the parsers under test has a corresponding evaluation on labelled F-score in the English parsing literature, trained on English CCGbank. However, despite the fact that the three English experiments precisely parallel the three Chinese experiments in methodology, we observed several striking differences.

Firstly, while a gap of only 0.2% separated English CCG parsing results on the P&K and C&C parsers, a significantly larger deficit of 4.8% separates the same parsers trained on Chinese CCGbank. Secondly, while Zhang and Clark (2011) were able to achieve figures roughly comparable with the state-of-the-art in C&C (Clark and Curran, 2007b) with a much simpler shift-reduce parser, the corresponding parser trained on Chinese CCGbank improves on the P&K model by 0.4%, and on the C&C model by a full 5.2%.

Our results confirm that not only does the gap between Chinese and English results on treebank parsing also exist in CCGbank-style parsing, the performance characteristics of three parsers on Chinese data differs dramatically from the that of the corresponding English parsers. Furthermore, the analysis in Figure 6.5a after McDonald and Nivre (2007)
which graphs dependency length against labelled F-score reveals some unexpected characteristics. Both the P&K and Z&C parsers (the blue line and the green line respectively) outperform the C&C parser in very short-range dependencies (spans of 1–9 words). However, the ability of both Z&C and C&C to correctly resolve longer-range dependencies starts to lag behind that of P&K in the short range (>) 5 words) and beyond, a difference which is not at all obvious from the absolute numeric evaluation in Table 6.12. This suggests that Z&C may be attaching the more frequent, shorter-range dependencies correctly at a higher rate than C&C and P&K, while the P&K model is better equipped to resolve long-range dependencies.

The total number of dependencies retrieved also differs between the three parsers (Figure 6.5b). All parsers under test retrieve similar numbers of short-range dependencies, but the difference becomes pronounced as the dependency length increases. While C&C retrieves the most medium- to long-range dependencies (> 20 words), it does so with low precision and recall (Figure 6.6). At this range, Z&C and C&C perform similarly, an unexpected result, while P&K maintains high accuracy regardless of dependency length. Finally, Z&C proposes very long-range dependencies (> 40 words) less often than the other parsers, which may highlight a bias in its model. This is also consistent with the low recall of Z&C on very long-range dependencies seen in Figure 6.6.

The intersection set experiment in Table 6.11 also illuminates differences between the three parsers. Evaluating only over sentences parsed by both Z&C and C&C (Z&C ∩ C&C), C&C is boosted by 0.25%, and Z&C by 0.31%. However, evaluating over the coverage intersection of P&K and C&C (P&K ∩ C&C), while P&K is boosted by only 0.22%, C&C is boosted by a larger 1.25%. This suggests that the three parsers are experiencing coverage loss on different types of sentences, a line of inquiry which we pursue later.

Finally, we note that the random initialisation of P&K has a clear effect on the pair of resulting models, with the default initialisation I − 6 outperforming the second seed I − 6* by 0.32% (Table 6.12). This suggests that the product parser approach of Petrov (2010) may buy further gains on Chinese CCGbank as well.

6.12 Summary

To conclude our exposition of parsing results, we note that while these results validate the thesis of this work — that efficient, wide-coverage Chinese ccg is possible with an automatically extracted corpus — performance lags considerably compared to English results, consistent with comparisons between treebank parsers. We have uncovered striking dif-
ferences between the behaviour of three parsers on English data, and the behaviour of the same parsers in parallel experiments on Chinese CCGbank.

The experiments which conclude this chapter aim to resolve what characteristics of Chinese result in the observed behaviour, and also aim to expose further the differences between the ability of three parsers to retrieve the NLDs which were the focus of our earlier analysis in developing Chinese CCGbank.
Chapter 7

The challenges of parsing Chinese CCG

白日依山盡 黃河入海流
欲窮千里目 更上一層樓
The sun sets behind the mountains.
The Yellow River flows seaward.
But if you seek a better sight,
you must climb another flight.

《登鸛雀樓》 Climbing Stork Tower

Having obtained the first Chinese CCG parsing results in a multi-parser evaluation over three state-of-the-art parsers, we investigate the characteristics of our corpus and of Chinese which cause the observed deficit.

In this chapter, we introduce the parser-aided corpus conversion methodology, which allows us to investigate the impact of particular corpus annotation decisions by generating variants of the original corpus, then training and evaluating a parser on the resulting corpora. For instance, we used the parser-aided corpus conversion methodology to conclude that the bare/non-bare NP distinction present in English CCGbank leads to undesirable lexical and derivational ambiguity in a parser, and therefore reject it not only on the theoretical grounds discussed in Chapter 2 but also on practical grounds.

We then investigate the coverage loss behaviour of the three parsers, observing that in Z&C and P&K, the two parsers which rely on the C&C tool generate to produce gold-standard dependencies, the majority of coverage losses are generated by the parser itself and not by this post-processing step.

We investigated the role that POS tagger and supertagger error played in the C&C parser, leading to inferior performance compared to the other two parsers. For the POS tagger, we quantify the impact of particular tagging errors by introducing particular confusion...
Chapter 7. The challenges of parsing Chinese CCG

types one at a time, allowing us to isolate the impact of each confusion. We find that the familiar Chinese noun/verb ambiguity, when decided incorrectly, is responsible for a large degradation in parser performance. In the C&C supertagger, we demonstrate by analysing the resulting false positive errors, that Chinese supertagging and parsing are difficult even with correct POS tags.

Finally, we focus on the performance of the three parsers on the NLD types which are a central challenge not only of Chinese parsing, but of the abstract CCG analysis and corpus conversion described in this dissertation. We show that no single parser performs well recovering all the NLD types we annotate for.

The generation of Chinese CCGbank described in this dissertation is only the first step in realising deep, accurate wide-coverage Chinese parsing. It is clear that properties of Chinese syntax, and of our CCG representation of Chinese syntax, pose considerable challenges for the accurate retrieval of dependencies. This chapter is the first step to understanding why.

7.1 The parser-aided corpus conversion methodology

As we saw in Chapters 2 and 3, the act of crafting analyses in a given formalism involves tradeoffs between competing concerns such as ease of annotation, the domain of the corpus, fidelity to the base formalism, and the intended application of the corpus. Often, as with our analysis of V-O compounds in Chapter 2, and the pseudo-cleft construction in Chapter 3, a prominent constraint is whether the annotation in the source corpus facilitates the retrieval of a given analysis. For instance, because the Tsinghua Chinese Treebank corpus lacks any annotation for extraction (Zhou, 1997), our extraction analysis of Section 3.2 would be difficult to obtain without costly manual annotation or heuristics.

In many cases, we presented a number of competing analyses for a given construction which exercised different tradeoffs, but were eventually forced to choose one particular analysis, often on the basis of the construction’s distribution in a corpus, or the effect of the annotation decision on ambiguity. In this section, we argue that there is no reason why the result of an automatic corpus conversion must embody one and only one set of annotation decisions. This simple idea has two particular applications. Firstly, different applications may wish to exercise different compromises — for instance, the bare/non-bare NP distinction considered in Chapter 2 involves a tradeoff between semantic transparency (the ability to distinguish common nouns ⟨e, t⟩ from proper nouns e) and lexical ambiguity (the proliferation of categories mentioning both N and NP). An application, such as pars-
The second application is a new methodology of integrating a parser into the corpus development process, which we call parser-aided corpus conversion.

This methodology, used with a parser on which the candidate corpora can be trained, can be used to guide the development of annotation decisions. Consider the bare/non-bare NP distinction again. While lexical ambiguity can be measured as a static property of the resulting corpus (say, by computing the average number of categories per word), derivational ambiguity — the productivity of a given type-change rule — is difficult to quantify without parsing. In this section, we consider varying two annotation decisions in the corpus, and evaluating the resulting changes in lexical ambiguity (a property of the corpus) and derivational ambiguity (a property of parsers trained on the corpus).

We apply the parser-aided corpus conversion methodology to produce several versions of Chinese CCGbank, to exercise tradeoffs between categorial ambiguity and derivational ambiguity which arise when representing aspects of Chinese syntax in the CCG setting. We then evaluate the impact of these changes by parsing the resulting variants of Chinese CCGbank with two of the parsers tested earlier in the multi-parser empirical evaluation: C&C (Clark and Curran, 2007b) and P&K (Petrov and Klein, 2007).¹

Because we used the parser-aided corpus conversion methodology during the development of Chinese CCGbank itself, these experiments reflect an earlier version of Chinese CCGbank. This earlier version makes the N-NP distinction which we described in Section 2.4.2. Accordingly, the parsing figures are not directly comparable with those of the previous chapter.

### 7.1.1 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.

¹ The experiments in this section only compare C&C and P&K, as the performance deficit between P&K and Z&C is much smaller by comparison (Table 6.12).
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 NPs.

(2)  a. 在 房子 里
    at room in:LC
    \[
    \frac{PP/LCP}{NP} \frac{LCP/NP}{LCP} \!
    \]
    \[
    \frac{PP}{NP} \frac{LCP/NP}{LCP \rightarrow NP} \!
    \]

b. 在 房子 里
    at room in:LC
    \[
    \frac{PP/NP}{NP} \frac{LCP/NP}{LCP \rightarrow NP} \!
    \]
    \[
    \frac{PP}{NP} \frac{LCP/NP}{LCP \rightarrow NP} \!
    \]

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.

Corpus Base, like (2-a), makes the distinction between categories LCP and NP. However, in corpus Unary-LCP, we test the impact of applying (2-b), in which the unary promotion LCP $\rightarrow$ NP is available.
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: \( N \rightarrow NP \). We present evidence in Section 2.4.2 that because bare/non-bare marking on NPs is not obligatory in Chinese, and because the determiner does not close off NP as it does in English, the \( N-NP \) distinction may not be a useful one in a Chinese CCG grammar.

We evaluated the impact of the bare/non-bare distinction on parsing models trained on the resulting corpora by creating a version of Chinese CCGbank (corpus No-Bare-N) which neutralises the distinction. This eliminates the atomic category \( N \), as well as the promotion rule \( N \rightarrow NP \).

### 7.1.2 Evaluating static properties of the corpus

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 7.1. Decomposing the category inventory 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.

### 7.1.3 Results

We follow the same experimental setup used in the multi-parser empirical evaluation in Section 6.10. The data splits, parser settings for the two parsers under test, and the same
Chapter 7. The challenges of parsing Chinese CCG

<table>
<thead>
<tr>
<th>model</th>
<th>LF</th>
<th>Lsa %</th>
<th>stag</th>
<th>cov</th>
<th>log C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I-3</td>
<td>68.97</td>
<td>13.45</td>
<td>83.64</td>
<td>95.7</td>
<td>-</td>
</tr>
<tr>
<td>I-6</td>
<td>71.67</td>
<td>15.70</td>
<td>85.00</td>
<td>96.4</td>
<td>-</td>
</tr>
<tr>
<td>GOLD</td>
<td>75.45</td>
<td>16.70</td>
<td>89.43</td>
<td>99.4</td>
<td>14.55</td>
</tr>
<tr>
<td>AUTO</td>
<td>66.32</td>
<td>12.81</td>
<td>83.88</td>
<td>98.6</td>
<td>14.69</td>
</tr>
<tr>
<td>Unary-LCP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I-3</td>
<td>69.75</td>
<td>14.15</td>
<td>84.07</td>
<td>96.0</td>
<td>-</td>
</tr>
<tr>
<td>I-5</td>
<td>71.40</td>
<td>14.83</td>
<td>84.97</td>
<td>96.4</td>
<td>-</td>
</tr>
<tr>
<td>GOLD</td>
<td>75.41</td>
<td>16.67</td>
<td>89.50</td>
<td>99.6</td>
<td>14.74</td>
</tr>
<tr>
<td>AUTO</td>
<td>66.24</td>
<td>12.61</td>
<td>83.95</td>
<td>98.7</td>
<td>14.75</td>
</tr>
<tr>
<td>No-Bare-N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I-3</td>
<td>70.22</td>
<td>16.49</td>
<td>84.37</td>
<td>96.5</td>
<td>-</td>
</tr>
<tr>
<td>I-5</td>
<td>72.74</td>
<td>18.59</td>
<td>85.61</td>
<td>96.5</td>
<td>-</td>
</tr>
<tr>
<td>GOLD</td>
<td>76.73</td>
<td>20.56</td>
<td>89.66</td>
<td>99.5</td>
<td>13.58</td>
</tr>
<tr>
<td>AUTO</td>
<td>66.95</td>
<td>14.62</td>
<td>83.90</td>
<td>99.2</td>
<td>13.86</td>
</tr>
</tbody>
</table>

Table 7.2: Dev set evaluation for P&K and C&C on the three corpus variations

<table>
<thead>
<tr>
<th>model</th>
<th>LF</th>
<th>Lsa %</th>
<th>stag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I-6</td>
<td>71.74</td>
<td>15.87</td>
<td>85.29</td>
</tr>
<tr>
<td>AUTO</td>
<td>67.50</td>
<td>15.36</td>
<td>84.52</td>
</tr>
<tr>
<td>Unary-LCP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I-5</td>
<td>71.40</td>
<td>14.97</td>
<td>85.26</td>
</tr>
<tr>
<td>AUTO</td>
<td>67.72</td>
<td>14.97</td>
<td>84.68</td>
</tr>
<tr>
<td>No-Bare-N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I-5</td>
<td>72.84</td>
<td>18.69</td>
<td>86.04</td>
</tr>
<tr>
<td>AUTO</td>
<td>68.43</td>
<td>16.17</td>
<td>84.57</td>
</tr>
</tbody>
</table>

Table 7.3: Dev set evaluation for P&K and C&C on PCTB 6 sentences parsed by both parsers

evaluation metrics and procedure are used in these experiments. In the experiments, log C indicates the logarithm of the C&C chart size, and allows us to quantify parser ambiguity.

Table 7.2 shows the performance of P&K and C&C on the three dev sets, and Table 7.3 only over sentences parsed by both parsers. For P&K on corpus Base, F-score and supertagger accuracy increase monotonically as further split-merge iterations refine the model. P&K on Unary-LCP and No-Bare-N overfits at 6 iterations, consistent with Fowler and Penn’s findings for English.
Table 7.4: Test set evaluation for P&K and C&C on corpus **No-Bare-N**

<table>
<thead>
<tr>
<th>model</th>
<th>LF</th>
<th>Lsa %</th>
<th>stag</th>
<th>cov</th>
<th>log C</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-5</td>
<td>72.73</td>
<td>20.28</td>
<td>85.43</td>
<td>97.1</td>
<td>-</td>
</tr>
<tr>
<td><strong>No-Bare-N</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GOLD</td>
<td>76.89</td>
<td>22.90</td>
<td>89.63</td>
<td>99.1</td>
<td>14.53</td>
</tr>
<tr>
<td>AUTO</td>
<td>67.09</td>
<td>15.28</td>
<td>83.95</td>
<td>98.7</td>
<td>14.89</td>
</tr>
</tbody>
</table>

The ~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 7.3, 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 showing the impact of incorrect **POS** tags.

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

Comparing **Base** and **Unary-LCP** in Table 7.2, C&C receives small increases in supertagger accuracy and coverage, but parsing performance remains largely unchanged; P&K performance degrades slightly. On both parsers, **No-Bare-N** 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 **No-Bare-N** 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 7.3, 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 7.4 shows that the behaviour of both parsers on the test section is consistent with the dev section.

### 7.1.4 Corpus ambiguity

To understand why corpus **No-Bare-N** is superior for parsing, we compare the ambiguity and sparsity characteristics of the three corpora. Examining log C, the average log-chart size (Table 7.2) shows that the corpus **Unary-LCP** changes (the addition of the unary rule  

\( LCP \rightarrow NP \)) increase ambiguity, while the additional corpus **No-Bare-N** changes (elimin-
Chapter 7. The challenges of parsing Chinese CCG

<table>
<thead>
<tr>
<th>corpus</th>
<th>Avg. tags/word</th>
<th>Grammar size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>1.84</td>
<td>1177</td>
</tr>
<tr>
<td>Unary-LCP</td>
<td>1.83</td>
<td>1084</td>
</tr>
<tr>
<td>No-Bare-N</td>
<td>1.79</td>
<td>964</td>
</tr>
</tbody>
</table>

Table 7.5: Corpus statistics

<table>
<thead>
<tr>
<th>corpus</th>
<th>V</th>
<th>P</th>
<th>M</th>
<th>N</th>
<th>O</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>791</td>
<td>158</td>
<td>56</td>
<td>2</td>
<td>170</td>
<td>1177</td>
</tr>
<tr>
<td>Unary-LCP</td>
<td>712</td>
<td>149</td>
<td>55</td>
<td>2</td>
<td>166</td>
<td>1084</td>
</tr>
<tr>
<td>No-Bare-N</td>
<td>670</td>
<td>119</td>
<td>41</td>
<td>1</td>
<td>133</td>
<td>964</td>
</tr>
</tbody>
</table>

Table 7.6: Grammar size, categorised by shape

ating the $N$-$NP$ distinction, resulting in the removal of the unary rule $N \rightarrow NP$) have the net effect of reducing ambiguity.

Table 7.5 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 7.2. While the size of the lexicon is reduced in Unary-LCP, the corresponding log $C$ figure in Table 7.2 increases slightly, because of the additional unary rule.

Table 7.6 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 $NP$.

7.1.5 Summary

On the basis of the parser-aided corpus conversion methodology, coupled with the theoretical justification in Section 2.4.2, we selected an analysis which collapses the $N$-$NP$ distinction for Chinese CCGbank.
7.2. Coverage loss behaviour for each parser

<table>
<thead>
<tr>
<th>Source of loss</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>C&amp;C incurred during training</td>
<td></td>
</tr>
</tbody>
</table>
*Gold-standard generation loss*  
Sentences are lost from the training data because the parser cannot reproduce the gold standard parses. Can be because of missing unary rules, binary rules or markedup entries. |
| C&C incurred during testing |  
*Explode*  
The size of the chart exceeded a configurable maximum during parsing.  
*No span*  
The parser could not find an analysis given any of the specified supertagger ambiguity levels $\beta$. |
| P&K |  
*No span*  
P&K could not obtain a spanning analysis, returning an empty tree. |
| Z&C |  
*Partial analysis*  
Z&C could not obtain a spanning analysis, but returned fragmentary parses (the partial derivations in its stack). |
| generate failures (applicable to P&K and Z&C) |  
*Parser generated incorrect rules*  
The parser produced derivations with incorrect rules, causing `generate` to reject the tree.  
*Unhandled rules*  
The parser produced derivations with rules not handled by `generate`, causing it to reject the tree. |

Table 7.7: Sources of coverage loss in each parser

The integration of parsers into the process of corpus conversion, to judge the viability of annotation choices, allows implementors of corpus conversions to understand the dynamic properties of the resulting corpora. Parser-aided corpus conversion will allow for the development of higher-quality corpora, by illuminating the trade-offs between efficiency, accuracy and formalistic purity which arise from choosing one analysis over another.
Chapter 7. The challenges of parsing Chinese CCG

7.2 Coverage loss behaviour for each parser

The multi-parser evaluation in the previous chapter confirmed that the three parsers under test possess vastly different coverage characteristics. But the conditions under which sentences can be lost differ in each parser. We show in this section that in parsers with complex architectures, a simple coverage statistic is not sufficiently nuanced; to understand coverage loss, we analyse the error conditions which can arise in the processing pipeline of each parser (Table 7.7).

During C&C parser training, the parser first attempts to reproduce each gold-standard derivation, by running the generate tool first described in Section 6.4. This runs the parser in a deterministic mode over the gold-standard derivations, so that parser features can be extracted for training. If the parser cannot reproduce the gold-standard derivation, the sentence is rejected. This can be due to missing rule coverage (rules which occur in the CCGbank but are not implemented in C&C), or missing entries in the markedup file.

During C&C parsing, each sentence in the test set is parsed at each ambiguity level $\beta$ in turn until a spanning analysis exists. Parsing can fail because the chart grows too large (explode), or else because all the ambiguity levels have been exhausted (no span), or a combination of these two.

The remaining two parsers have in common their dependence on the C&C tool generate, which is used to convert the output of the constituent parsers to CCG dependencies, operating in the same deterministic mode used during C&C parser training. We are careful to stress that there are two stages at which coverage loss can occur in P&K and Z&C: within the parser itself, or else within generate.

Parser-internal coverage loss results when the parser cannot find a spanning analysis, and outputs an empty tree (P&K) or a forest of partial derivations (Z&C). An empty tree is an immediate coverage loss, but the partial derivations output by Z&C can still be processed. As Zhang and Clark (2011) note, an advantage of shift-reduce parsing is that if no spanning analysis can be obtained, the partial trees in the stack can be returned in lieu of returning nothing at all. We modified generate to accept partial parses, and generate dependencies from the partial subtrees.\footnote{Z&C decomposes a partial parse into intact subtrees, and sends each subtree to generate separately (Yue Zhang, p.c.). Instead, we modified generate to accept the tree of partial derivations directly and only process the intact subtrees.}

The second stage at which coverage loss can occur in P&K and Z&C is when generate is run to extract dependencies from the derivations output by each parser. Coverage loss during generate comes in two varieties; parser-originated coverage loss, and generate-
ate-originated coverage loss. Parser-originated coverage loss occurs when the derivations contain incorrect rule combinations or categories not in the marked-up file.\(^3\) \textit{generate}-originated coverage loss, on the other hand, arises when the derivations output by the parser contain legitimate rules not implemented in \textit{generate}. This is the same class of error as the \textit{gold-standard generation loss} incurred by the C&C parser during training — rules which may occur in the data, but are not implemented in \textit{generate} and hence cause the sentence to be rejected.

Having decomposed the sources of coverage loss in each parser, we quantify the coverage loss of each type in each parser model in Table 7.8. While C&C loses 4.7\% of its training data to rules not implemented in the parser, its coverage during testing is very high.

However, both P&K and Z&C encounter significant coverage problems, with the P&K parser failing to return an analysis in 109 instances. Z&C fails to obtain a spanning analysis in 28.8\% of trees, but since Z&C is able to return partial parses, this is not the same as the dependency coverage figure. Furthermore, Z&C only produces one analysis which is rejected by \textit{generate}. Out of the analyses which P&K did return, \textit{generate} fails to induce dependencies in a small number of cases due to P&K derivations yielding unhandled, or incorrect rules. This demonstrates that the vast majority of coverage losses are due to non-analyses from the parser, rather than errors in \textit{generate}. The fact that the C&C parser only considers \texttt{ccg} rule combinations (and type-change rules), on the other hand, prevents it from proposing non-analyses, an advantage of using a parser which is aware of the structure of \texttt{ccg} rules.

This analysis indicates that the differences between the three parsers extend to their coverage behaviour on Chinese CCGbank.

\(^3\) This latter case (the production of categories outside the marked-up file) cannot occur in P&K or Z&C, because both operate under the “seen rules” restriction which constrains the set of derivable categories to those which occur in the training data. We mention this error case because a parser which actually applies all available combinatory rules could derive categories not in the training data.
Chapter 7. The challenges of parsing Chinese CCG

<table>
<thead>
<tr>
<th>Error type</th>
<th>Parser</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C&amp;C AUTO</td>
</tr>
<tr>
<td>Training failures (% are proportions of training data)</td>
<td></td>
</tr>
<tr>
<td>Missing unary rule</td>
<td>407 (1.8%)</td>
</tr>
<tr>
<td>Missing binary rule</td>
<td>654 (2.9%)</td>
</tr>
<tr>
<td>Parse failures</td>
<td></td>
</tr>
<tr>
<td>Explode</td>
<td>3 (0.3%)</td>
</tr>
<tr>
<td>No span/partial analysis</td>
<td>0 (0.0%)</td>
</tr>
<tr>
<td>generate failures</td>
<td></td>
</tr>
<tr>
<td>Parser generated incorrect rules</td>
<td>-</td>
</tr>
<tr>
<td>Unhandled rules</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 7.8: Coverage losses for each parser (number of sentences)

7.3 Corrupting POS tags to isolate confusions

The performance deficit between the two configurations GOLD (gold-standard POS tags) and AUTO (automatic POS tags) of the C&C parser in Table 6.12 suggest that tagger accuracy plays an overwhelming part in the accuracy of Chinese parsers. However, it is not clear which particular classes of tagging errors are responsible. With a better understanding of which confusion types are the most deleterious to parsing accuracy, we could better direct future research in Chinese parsing.

Levy and Manning (2003) observed, for instance, that noun/verb ambiguity of the below type, can greatly affect parsing accuracy when resolved incorrectly:

(3)  设计  建设
design  build
design and construction
to design and build

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, we perform an experiment where we corrupt the gold POS tags, by gradu-
7.4. The effect of supertagger accuracy

<table>
<thead>
<tr>
<th>Confusion</th>
<th>LF</th>
<th>ΔLF</th>
<th>stag</th>
<th>cov</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base (GOLD)</td>
<td>76.10</td>
<td>89.41</td>
<td>99.55</td>
<td></td>
</tr>
<tr>
<td>NR ▷ NN</td>
<td>76.09</td>
<td>-0.01</td>
<td>89.41</td>
<td>99.55</td>
</tr>
<tr>
<td>JJ ▷ NN</td>
<td>76.00</td>
<td>-0.10</td>
<td>89.35</td>
<td>99.55</td>
</tr>
<tr>
<td>DEC ▷ DEG</td>
<td>74.38</td>
<td>-1.72</td>
<td>88.77</td>
<td>99.02</td>
</tr>
<tr>
<td>VV ▷ NN</td>
<td>73.03</td>
<td>-3.07</td>
<td>87.62</td>
<td>99.02</td>
</tr>
<tr>
<td>All (AUTO)</td>
<td>67.16</td>
<td>84.01</td>
<td>99.02</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.9: Corrupting C&C gold POS tags piecemeal on dev set of Chinese CCGbank. ΔLF is the change in LF when each additional confusion type is allowed.

ally re-introducing automatic POS errors on a cumulative basis, one confusion type at a time.4

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

Table 7.9 shows that while the confusion types NR ▷ NN and JJ ▷ NN have no impact on the evaluation, the confusions DEC ▷ DEG and VV ▷ NN, introduced cumulatively, cause reductions in F-score of 1.62 and 1.35% respectively. This is expected; the categories in Chinese CCGbank do 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 ▷ 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 observed a gap of 8.94% for Chinese.
Table 7.10: Analysis of the 8,511 false positive dependencies from C&C on PCTB 6 dev set

<table>
<thead>
<tr>
<th>correct pos</th>
<th>incorrect pos</th>
</tr>
</thead>
<tbody>
<tr>
<td>correct stag</td>
<td>2,339 (27.48%)</td>
</tr>
<tr>
<td>incorrect stag</td>
<td>4,598 (54.02%)</td>
</tr>
</tbody>
</table>

7.4 The effect of supertagger accuracy

Table 7.10 breaks down the 8,511 false positive dependencies 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, for example, attaching the incorrect argument to a head, or incorrectly deciding NP internal structure. The vast majority of false positives, though, are 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.

7.5 Analysis of parser accuracy by dependency type

Guo et al. (2007b) observed that while the recovery of NLDs is critical to the correct acquisition of predicate-argument structure, the vast majority of wide-coverage parsers simply do not retrieve them. They demonstrate that while this approach is tolerated in English, Chinese syntax generates NLDs more frequently than English, and a parser which does not retrieve them is much less useful.

The CCG literature is enriched by accounts of cross-linguistic non-local dependency types, including those arising from extraction and coordination phenomena (Steedman, 2000). The ability of parsers based on CCG to enact these analyses is one of the benefits of

---

4 Suppose the gold standard labels a given lexical item VV, and the automatic POS tagger labels it NN. If our experiment includes the confusion type VV $\Rightarrow$ NN, then we corrupt the gold standard by replacing the correct tag VV with the incorrect tag NN wherever the automatic POS tagger has made an error of this type.
the formalism. In Section 6.10, we trained and parsed Chinese CCGbank with three parsers, obtaining three models with vastly different performance characteristics. Figure 6.5 shows that the three parsers behave differently retrieving short- and long-distance dependencies, so a natural question is whether the differences in observed performance extend to their ability to retrieve different kinds of dependencies.

To determine how each parser performs in retrieving dependency types, we performed an experiment which considers the ability of each parser to retrieve particular dependency types only. Each row of Table 7.11 corresponds to a particular slot of a given category. For instance, we write the subject slot of the transitive verb category as \((S[dcl]\NP)/NP\), and its object slot as \((S[dcl]\NP)/NP\). For each slot, we report the number of dependencies which the parser retrieved for that slot (frequency) and labelled F-score.

**Recovering NP internal structure.** Z&C outperforms C&C and P&K attaching noun modifiers; P&K is inferior to the other two parsers attaching noun modifier modifiers.

**Verb argument attachment.** C&C underperforms the other two parsers attaching the arguments of intransitive and transitive verbs.

**Control/raising.** Of the three parsers, P&K attaches most accurately the local and non-local dependency arising from the control/raising construction.

**Extraction.** Surprisingly, Z&C is most accurate at determining the correct attachment level and clausal complement of subject extraction, but underperforms the other two parsers on object extraction. This suggests a bias in the model, which could uncover why shift-reduce parsers are effective for Chinese.

被 bei and 把 ba. There are relatively few instances of the passivisation constructions in the data. Z&C performs best recovering the dependencies arising from short 被 bei, but C&C performs best on long 被 bei dependencies. However, all three parsers return very few dependencies at all for long 被 bei. None of the three parsers return any dependencies for the gapped 把 ba construction.

The poor accuracy and recall on the 把 ba and 被 bei constructions is a consequence of their difficulty as well as their infrequency in the data. We suggest that the approach of Clark et al. (2004), who significantly increased parser accuracy on questions by manually annotating data with categories (but not derivation structure), may vastly increase the accuracy of Chinese CCGbank models on this rare class of NLD.

The NLD analysis of Table 7.11 shows that the summary statistics of Table 6.12 do not adequately characterise the performance of the three parsers. While Z&C obtains the highest labelled F-score of the three parsers, it does not perform uniformly well; it attaches short-range dependencies very accurately, but has poor recall and accuracy on longer-range de-
Chapter 7. The challenges of parsing Chinese ccg

dependencies. On the other hand, P&K suffers from coverage problems, but consistently maintains high accuracy regardless of dependency length, achieving the best results for object extraction and attaching verbal arguments. Meanwhile, C&C performs well on noun modifier attachment and short 被 bei, but underperforms on subject extraction.

It is clear that no single parser out of the three performs consistently well on all NLP types. The analysis in this section has uncovered intriguing differences between the three parsers, which we believe will shed light on the characteristics of Chinese which make it challenging to parse.

7.6 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 \( S[dcl] \backslash NP \rightarrow S[dcl] \), for which we surveyed three analyses present in the ccg literature in Section 3.5.1.

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:

\( a. \ t_i \ 出来 \ 的 \ 问题 \)
\( \text{t}_i \text{ come out DE question}_i \)
the questions which arise

\( b. \ pro \ 出来 \ 的 \ 问题 \)
\( \text{pro come out DE question} \)
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 7.12. 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.
<table>
<thead>
<tr>
<th>C&amp;C auto</th>
<th>P&amp;K I-6</th>
<th>Z&amp;C $\tau = 39$</th>
<th>category</th>
<th>NLD?</th>
<th>dependency function</th>
</tr>
</thead>
<tbody>
<tr>
<td>LF freq</td>
<td>LF freq</td>
<td>LF freq</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.78</td>
<td>4257</td>
<td>0.77</td>
<td>3679</td>
<td>0.84</td>
<td>4321</td>
</tr>
<tr>
<td>0.56</td>
<td>610</td>
<td>0.51</td>
<td>524</td>
<td>0.57</td>
<td>622</td>
</tr>
<tr>
<td>0.73</td>
<td>2167</td>
<td>0.80</td>
<td>2003</td>
<td>0.84</td>
<td>2229</td>
</tr>
<tr>
<td>0.65</td>
<td>1725</td>
<td>0.71</td>
<td>1608</td>
<td>0.71</td>
<td>1722</td>
</tr>
<tr>
<td>0.61</td>
<td>881</td>
<td>0.67</td>
<td>852</td>
<td>0.70</td>
<td>927</td>
</tr>
<tr>
<td>0.68</td>
<td>872</td>
<td>0.74</td>
<td>745</td>
<td>0.70</td>
<td>762</td>
</tr>
<tr>
<td>0.64</td>
<td>659</td>
<td>0.69</td>
<td>576</td>
<td>0.63</td>
<td>560</td>
</tr>
<tr>
<td>0.61</td>
<td>353</td>
<td>0.65</td>
<td>370</td>
<td>0.77</td>
<td>467</td>
</tr>
<tr>
<td>0.62</td>
<td>368</td>
<td>0.67</td>
<td>362</td>
<td>0.75</td>
<td>469</td>
</tr>
<tr>
<td>0.63</td>
<td>114</td>
<td>0.72</td>
<td>107</td>
<td>0.59</td>
<td>121</td>
</tr>
<tr>
<td>0.56</td>
<td>116</td>
<td>0.69</td>
<td>105</td>
<td>0.59</td>
<td>118</td>
</tr>
<tr>
<td>0.71</td>
<td>31</td>
<td>0.69</td>
<td>25</td>
<td>0.82</td>
<td>32</td>
</tr>
<tr>
<td>0.58</td>
<td>27</td>
<td>0.65</td>
<td>21</td>
<td>0.73</td>
<td>27</td>
</tr>
<tr>
<td>0.71</td>
<td>8</td>
<td>0.62</td>
<td>8</td>
<td>0.67</td>
<td>9</td>
</tr>
<tr>
<td>0.55</td>
<td>4</td>
<td>0.29</td>
<td>5</td>
<td>0.53</td>
<td>6</td>
</tr>
<tr>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7.11: Accuracy per dependency, for selected dependency types
Table 7.12: Dev set evaluation for C&C and P&K over pro-drop sentences only (and over full set in parentheses)

<table>
<thead>
<tr>
<th>model</th>
<th>LF</th>
<th>Lsa %</th>
<th>stag</th>
<th>cov</th>
</tr>
</thead>
<tbody>
<tr>
<td>GOLD</td>
<td>73.93</td>
<td>7.03</td>
<td>88.93</td>
<td>98.61</td>
</tr>
<tr>
<td></td>
<td>(76.08</td>
<td>19.35</td>
<td>89.41</td>
<td>99.47)</td>
</tr>
<tr>
<td>AUTO</td>
<td>64.99</td>
<td>4.28</td>
<td>83.48</td>
<td>97.23</td>
</tr>
<tr>
<td></td>
<td>(67.37</td>
<td>13.98</td>
<td>84.00</td>
<td>98.84)</td>
</tr>
<tr>
<td>I-6</td>
<td>69.95</td>
<td>8.29</td>
<td>84.82</td>
<td>96.64</td>
</tr>
<tr>
<td></td>
<td>(72.55</td>
<td>17.04</td>
<td>-</td>
<td>98.24)</td>
</tr>
</tbody>
</table>

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 \( C \) (chart size, a measure of parser ambiguity) 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.

### 7.7 Summary

The experiments in this chapter are the first investigation of what makes Chinese CCG difficult to parse. But to our knowledge, little work has focused on the characteristics of Chinese which make it a challenge to parse.

First, we demonstrated through the parser-aided corpus conversion methodology that the annotation decisions in the corpus affect the characteristics of parsers trained on the resulting corpora. We argue that corpus conversions can benefit from integrating parsers into the development cycle, as dynamic properties of the corpus, such as derivational ambiguity and the productivity of type-change rules, are best evaluated directly by training parser models on candidate corpora.

Second, through the POS corruption experiment, we have validated the observation of Levy and Manning (2003) that the Chinese noun/verb ambiguity contributes significantly to the difficulty of parsing Chinese, an error class compounded by the pipeline architecture of the C&C parser. This suggests a focus for improving the quality of Chinese parsers.
Third, we have shown that even with correct POS tags, the supertagger is still faced with a challenging task. This suggests that a joint supertagging/parsing model, such as Auli and Lopez (2011), may yield considerable gains in Chinese.

Finally, we decomposed the parser evaluation of Chapter 6 into the NLĐ types which were a focus of our abstract analysis, and subsequently our corpus conversion. We show that the summary statistics of the previous chapter hide subtle differences in the performance characteristics of the three parsers.

This point marks the culmination of our experiments. We have demonstrated, through Chinese CCGbank and a trio of Chinese CCG parsing models, the viability of deep, efficient Chinese CCG parsing, backed by a formalism which fully accounts for the non-local dependency types frequently generated by its syntax. With our resources as a testbed, researchers are now ready to begin a deeper investigation into the challenges of parsing Chinese.
Chapter 8

Conclusion

書山有路勤為徑，學海無涯苦作舟。
The scholar's road is diligence; on the shoreless sea of scholarship, toil is our vessel.

Han Yu (韓愈; 768–824)

We have successfully synthesised existing work in corpus conversion, Chinese syntax and parsing, wide-coverage CCG parsing, including recent results adapting non-CCG parsers to CCG, to produce and evaluate Chinese CCGbank, the first Chinese CCG corpus in the literature. This allowed us to experimentally validate the feasibility of wide-coverage Chinese CCG parsing. While observing a deficit between Chinese and English CCG parsing results which echoes the gulf between results in phrase structure parsing, we uncovered vast differences between the performance of three CCG parsers, all of which perform similarly in English. This suggests that CCG parsing in the two languages has different characteristics, calling for different parsers and NLP techniques.

8.1 Looking back

Chapter 1 traces the descent of Combinatory Categorial Grammar from the pure categorial grammar of Ajdukiewicz (1935) and Bar-Hillel (1953), presenting the problems in the AB grammar to which CCG was developed as a solution. We formally presented categories and combinatory rules, the power of the transparent syntax-semantics interface, and presented modern extensions to the CCG such as CCGbank-style grammars, multi-modal
Chapter 8. Conclusion

CCG and hat-CCG. Finally, we introduced several NLP applications of CCG which harness its abilities.

Chapter 2 developed the first account of Chinese grammar through CCG, harmonising existing linguistic analyses with the CCG setting. We justified the addition of new atomic categories to the Chinese CCGbank grammar, and argue on the basis of existing analyses of Chinese NP that the bare/non-bare distinction should be abandoned as it introduces considerable modifier ambiguity. We also argued that particular verb compounding strategies prevalent in Chinese are best handled as morphologically pre-composed units opaque to syntax, in contrast with their representation in the Penn Chinese Treebank, and contributed a CCG account of basic Chinese syntax.

Chapter 3 focused on harnessing the power of CCG to account for the challenging non-local dependencies which arise with great frequency in Chinese text. In this section we exploited existing analyses for NLD syntax from the CCG literature and created new ones, adapting the standard CCG analysis of extraction to Chinese, accounting for island effects and the additional configurations of the relative clause construction. We provided an analysis of topicalisation which reflects its greater frequency in Chinese compared to English, and developed an account of the 把/被 ba/bei constructions which harmonises accounts from the generative syntax literature. Finally, we presented a new analysis of Chinese wh-in-situ and successfully captured a wh-word asymmetry in the CCG setting.

Chapter 4 introduced the key resources which inform our corpus conversion approach: English CCGbank and the Penn Chinese Treebank. We demonstrated the characteristics of the Penn Chinese Treebank annotation through which Xue et al. (2005) encode gold-standard head structure, and show that this can eliminate a source of error in the corpus conversion process.

We then presented the framework of the Chinese CCGbank conversion algorithm, showing that these gold-standard head annotations can be incorporated into a corpus conversion.

Chapter 5 describes how we extract the analyses for NLD syntax previously developed in Chapter 3 from the Penn Chinese Treebank annotation. We exploited the power of a language which pattern-matches against treebank trees to identify instances of special syntax, and demonstrated how to reshape these into our abstract analyses. Finally, we analysed the static properties of Chinese CCGbank, and performed a manual evaluation of our silver-standard NLD extraction algorithm, showing that it correctly extracts the long distance dependency in 90% of manually annotated NLD instances.
Chapter 6 augments the CCG parsing literature with accounts of three Chinese parsing models, which precisely parallel the English results in Clark and Curran (2007b), Fowler and Penn (2010) and Zhang and Clark (2011). We evaluated improvements to the C&C pos- and supertagger, and obtained the surprising result that while all three parsers exhibited very similar performance trained on English CCGbank data, the corresponding Chinese parsers vary in performance by 5.2%. This suggested that Chinese parsing has different characteristics to English parsing, prompting further experiments.

Chapter 7 investigates what properties of Chinese syntax, and our encoding of Chinese syntax in Chinese CCGbank, are responsible for the performance gap. We introduced a new methodology, parser-aided corpus conversion, which evaluates the impact of annotation decisions by parsing on the resulting corpora. Based on this methodology, we chose to eliminate the bare/non-bare NP distinction first introduced in Chapter 2.

We also evaluated the role of tagger error in Chinese parser performance, and quantified the impact of particular constructions on the difficulty of Chinese parsing.

8.2 Looking forward

The groundbreaking development of English CCGbank by Hockenmaier (2003) directly enabled a productive line of research in wide-coverage CCG parsing which continues to this day.

The most prominent product of this dissertation is Chinese CCGbank itself, and the most exciting prospect is that the same techniques and applications unlocked by English CCGbank can now be investigated for Chinese. Chapter 6 has already showed that techniques which did not yield dramatic performance improvements in one language may yet prove valuable in another, and conversely that parser architectures such as the tagger/parser pipeline which are effective for English may not be appropriate for Chinese.

The analysis of Chapter 7 tries to answer the question: what characteristics of Chinese are responsible for the observed deficit in parser performance? We believe that a single summary statistic cannot answer this question, and yet the majority of research in parsing stops at the production of a figure.

Some researchers, however, have investigated decomposing parsing errors to better understand the error conditions which exist, and their distribution. Kummerfeld et al. (2012) uses a greedy search which iteratively transforms an automatic parse into the gold-standard parse by applying tree surgery operations. The number of tree moves of each type which
must be applied to transform the candidate sentences into the gold standard thus quantify and categorise the error cases in the data.

The error categorisation approach is likely to lead to a better understanding of the error types which occur in Chinese, and whether particular error types familiar in English are rare in Chinese (or vice versa). This has the potential to direct the future of research in Chinese parsing.

Another result from Chapter 7 is that particular confusion types of Chinese tagger error, such as the verb-noun confusion, are particularly deleterious to parsing accuracy. This is compounded by the pipeline architecture of C&C, because a mis-determination in the POS tagger, very early in the parsing process, is difficult to subsequently recover from.

Given the difficulty of deciding the verb/noun ambiguity locally, this suggests that joint tagging and parsing approaches, with which Auli and Lopez (2011) have shown gains in parsing accuracy in English, may in fact yield much larger gains in Chinese.

A third direction stems from our work in Section 7.1, in which we vary the annotation decisions in the corpus to evaluate their effect on resulting tagging and parsing models. As we noted in Section 5.2, due to the properties of the source annotation, the actual realisations in Chinese CCGbank of certain constructions differ from our chosen analyses in the abstract grammar of Chapters 2 and 3. Extracting these should allow parsers to generalise better over the data.

8.3 Conclusion

We have augmented the literature with CCG analyses of Chinese syntax, developed a new CCG bank which brings the benefits of CCG to Chinese, and obtained three wide-coverage parsing models with different characteristics. While it is clear that Chinese CCG parsing involves significant challenges, we have opened the task up to empirical investigation.
Appendix A

Unary projections in the Penn Chinese Treebank

Table A.1 lists the unary projections which are shrunk by the Binarise algorithm (§4.6.3). The notation $X < Y$ indicates that the algorithm should apply the shrink operation (replacing the parent with its child) when a node with tag $X$ is a unary projection of $Y$.

$N_X$ represents the set of Penn Chinese Treebank noun tags: \{NN, NT, NR\}, and $V_X$ represents the verb tag set: \{VV, VA, VC, VE, VPT, VSB, VRD, VCD, VNV\}.

Greyed out tags in the pattern result from instances of systematic annotation errors, which we nevertheless collapse to improve data quality.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP $&lt;$ $N_X$</td>
<td>QP $&lt;$ M</td>
</tr>
<tr>
<td>VP $&lt;$ Vx</td>
<td>AD</td>
</tr>
<tr>
<td>ADJP $&lt;$ JJ</td>
<td>AD</td>
</tr>
<tr>
<td>ADVP $&lt;$ AD</td>
<td>CS</td>
</tr>
<tr>
<td>NP-MNR</td>
<td>NP-PRP $&lt;$ Nx</td>
</tr>
<tr>
<td>NP-PN $&lt;$ NR</td>
<td>PRN $&lt;$ PU</td>
</tr>
<tr>
<td>CLP $&lt;$ M</td>
<td>LST $&lt;$ PU</td>
</tr>
<tr>
<td>LCP $&lt;$ LC</td>
<td>DNP $&lt;$ QP</td>
</tr>
<tr>
<td>DP $&lt;$ DT</td>
<td>OD</td>
</tr>
<tr>
<td>FLR</td>
<td>FW $&lt;$ *</td>
</tr>
</tbody>
</table>

Table A.1: Unary projections
Appendix B

Type-change rules in Chinese CCGbank

We summarise the type-change rules implemented in Chinese CCGbank.

<table>
<thead>
<tr>
<th>Description</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject pro-drop</td>
<td>$S[\text{dcl}] \mapsto S[\text{dcl}]$ (§3.5.1)</td>
</tr>
<tr>
<td>Object pro-drop</td>
<td>$(S[S]/NP)/NP \mapsto S[S]/NP$</td>
</tr>
<tr>
<td></td>
<td>$(S/S)/NP \mapsto S/S$</td>
</tr>
<tr>
<td></td>
<td>$(VP/VP)/NP \mapsto VP/VP$</td>
</tr>
<tr>
<td>Elision of numeral in Num + MW</td>
<td>$M \mapsto NP/NP$ (§2.4)</td>
</tr>
<tr>
<td>Null relativiser extraction</td>
<td>$S[\text{dcl}]/NP \mapsto NP/*$ (§3.2.2)</td>
</tr>
<tr>
<td>Non-gapped topicalisation</td>
<td>$(T_{\text{ngap}}) \quad XP \mapsto S/S$ (§3.3) where $XP \in {NP, QP, S[\text{dcl}]}$</td>
</tr>
<tr>
<td>Gapped topicalisation</td>
<td>$(T_{\text{gap}}) \quad XP \mapsto S/(S/XP)$ (§3.3) where $XP \in {NP, QP, S[\text{dcl}]}$</td>
</tr>
</tbody>
</table>

Figure B.1: Unary type-change rules in Chinese CCGbank
### Appendix B. Type-change rules in Chinese CCGbank

#### 1. Description

<table>
<thead>
<tr>
<th>Description</th>
<th>NP</th>
<th>NP</th>
<th>NP-NP apposition (§2.4.4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP-NP</td>
<td>NP</td>
<td>NP</td>
<td>apposition (§2.4.4)</td>
</tr>
</tbody>
</table>

**Unlike coordination rules** (down to threshold 5 in PCTB; §2.6.5)

| conj NP    | S[|dcl|][conj] |
|------------|--------------|
| conj S[|dcl|] NP | NP[conj]     |
| conj NP    | (NP/NP)[conj] |
| conj S[|dcl|] NP | (S[|dcl|]\NP)[conj] |
| conj S[|dcl|]\NP | NP[conj]     |

**Figure B.2: Binary type-change rules in Chinese CCGbank**

<table>
<thead>
<tr>
<th>Description</th>
<th>NP</th>
<th>T/(T/NP) T ∈ {S}</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCP</td>
<td>T/(T/LCP) T ∈ {S}</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NP</th>
<th>T/(T/NP) T ∈ {VP, TV, VP/PP, VP/QP, VP/VP}</th>
</tr>
</thead>
<tbody>
<tr>
<td>S[</td>
<td>dcl</td>
</tr>
<tr>
<td>QP</td>
<td>T/(T/QP) T ∈ {VP}</td>
</tr>
<tr>
<td>VP</td>
<td>T/(T/VP) T ∈ {VP}</td>
</tr>
</tbody>
</table>

**Figure B.3: Licensed type-raising operations in the Chinese CCGbank lexicon**
Appendix C

Fixes to PCTB annotation

This appendix describes the changes we made to the PCTB annotation to account for systematic inconsistencies.

<table>
<thead>
<tr>
<th>Fix</th>
<th>Schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hoisting paired quotes</td>
<td><img src="image1.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Missing NP projection</td>
<td><img src="image2.png" alt="Diagram" /></td>
</tr>
<tr>
<td>ḡ ḡ de-nominalisation has NP function</td>
<td><img src="image3.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Missing -OBJ tag</td>
<td><img src="image4.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>
Fix | Schema
---|---
Removing repeated unary projections | \[IP|VP|NP|CP \rightarrow IP|VP|NP|CP\]
| \[IP|VP|NP|CP\]

**DP-SBJ \rightarrow QP-SBJ**

| XP \rightarrow XP |
| XP PU PRN \rightarrow XP PRN |

**DEG in place of DEC**

| CP \rightarrow CP |
| IP DEG \rightarrow IP DEC |

**Missing NP projection**

| NP \rightarrow NP |
| CP NP-APP NP-PN \rightarrow CP NP |

**Missing phrasal projection in ADVP modification**

| IP \rightarrow IP |
| NP-SBJ ADVP VP \rightarrow NP-SBJ VP |

**Missing phrasal projection NP**

| DNP \rightarrow DNP |
| PN DEG \rightarrow NP DEG |

---
<table>
<thead>
<tr>
<th>Fix</th>
<th>Schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>Re-insertion of <em>pro</em></td>
<td>IP → IP</td>
</tr>
<tr>
<td></td>
<td>VP → NP-SBJ VP</td>
</tr>
<tr>
<td></td>
<td><em>pro</em></td>
</tr>
<tr>
<td>Mis-tagging ADVP</td>
<td>NP → JJ</td>
</tr>
<tr>
<td></td>
<td>JJ → JJ</td>
</tr>
<tr>
<td>Mis-tagging NP → VV</td>
<td>NP → VP</td>
</tr>
<tr>
<td></td>
<td>VP</td>
</tr>
<tr>
<td>Extraneous projection NP → ADJP</td>
<td>NP → ADJP</td>
</tr>
<tr>
<td></td>
<td>ADJP</td>
</tr>
<tr>
<td></td>
<td>JJ</td>
</tr>
<tr>
<td>Extraneous projection NP → QP</td>
<td>NP → NP</td>
</tr>
<tr>
<td></td>
<td>QP</td>
</tr>
<tr>
<td></td>
<td>QP</td>
</tr>
<tr>
<td>Mis-tagging CLP → NN</td>
<td>CP-APP → CP-APP</td>
</tr>
<tr>
<td></td>
<td>IP</td>
</tr>
</tbody>
</table>

293
## Fix

<table>
<thead>
<tr>
<th>Fix</th>
<th>Schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mis-tagging <strong>CLP</strong> → <strong>NN</strong></td>
<td><strong>CLP</strong> → <strong>CLP</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>NN</strong> → <strong>M</strong></td>
</tr>
<tr>
<td>Mis-tagging <strong>VP</strong> → <strong>NN</strong></td>
<td><strong>VP</strong> → <strong>VP</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>NN</strong> → <strong>VV</strong></td>
</tr>
<tr>
<td><strong>Treat</strong> <em>shi…de</em> as <strong>NP</strong> headed by <em>de</em></td>
<td><strong>NP-PRD</strong> → <strong>NP-PRD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>CP</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Remove superfluous projection marking parenthetical</strong></td>
<td><strong>PRN</strong> → **NP</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>**NP</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Treat</strong> <strong>CS</strong> (subordinator) as head</td>
<td><strong>ADVP</strong> → <strong>CS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>CS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Mis-tagging</strong> <strong>CP</strong> → <strong>M</strong></td>
<td><strong>CP</strong> → <strong>M</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>M</strong></td>
</tr>
<tr>
<td><strong>Reshaping long 被 bei construction</strong></td>
<td><strong>LB</strong> * → <strong>LB</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>NP-SBJ</strong> VP</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Fix</td>
<td>Schema</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>Missing NP projection in DNP</td>
<td>![Diagram of NP projection in DNP]</td>
</tr>
<tr>
<td>Missing LCP projection</td>
<td>![Diagram of LCP projection]</td>
</tr>
<tr>
<td>Wrong attachment of DEC</td>
<td>![Diagram of DEC attachment]</td>
</tr>
<tr>
<td>Treating IP-TPC as VP-type sentential pre-modifier</td>
<td>![Diagram of IP-TPC as VP]</td>
</tr>
<tr>
<td>Reshaping PRO-gapped IP complements</td>
<td>![Diagram of PRO-gapped IP complements]</td>
</tr>
</tbody>
</table>
Appendix C. Fixes to RSTB annotation

<table>
<thead>
<tr>
<th>Fix</th>
<th>Schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reshaping IP complements</td>
<td><img src="" alt="Diagram" /></td>
</tr>
</tbody>
</table>
Bibliography


