Recovering Chinese Nonlocal Dependencies with a Generalized Categorial Grammar

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Manjuan Duan, MA, MS

Graduate Program in Department of Linguistics

The Ohio State University

2019

Dissertation Committee:

William Schuler, Co-Advisor
Carl Pollard, Co-Advisor
Micha Elsner
Zhiguo Xie
Abstract

Correctly resolving nonlocal dependencies is challenging (Rimell et al., 2009) but crucial for natural language processing tasks such as question-answering and information extraction. It is especially important for Mandarin Chinese, which makes a heavy use of nonlocal dependencies (Kummerfeld et al., 2013). This thesis describes a Generalized Categorial Grammar (GCG) which has a slightly larger set of inference rules tailored specifically for Mandarin Chinese and a much smaller set of categories compared with other more lexicalized categorial grammars. The -g category in the grammar provides an intuitive account for the filler-gap constructions in Mandarin Chinese.

The thesis introduces a systematic reannotating process which converts the Penn Chinese Treebank annotations (Xue et al., 2005) into GCG annotations. The resulting broad-coverage GCG annotations can be used to train a probabilistic grammar using an automatic constituent parser. The reannotation process takes full advantages of the trace information annotated in the Penn Chinese Treebank so that the resulting GCG annotations retain the information needed for resolving nonlocal dependencies.

A syntactic parsing evaluation shows that a Berkeley latent-variable parser (Petrov and Klein, 2007) trained on the Chinese GCG annotations is significantly more accurate than the same parser trained on a more lexicalized categorial grammar in predicting a common test set of gold binary unlabeled trees in both grammars. A semantic dependency parsing system based on GCG syntactic features achieves comparable performance for the overall labeled dependency prediction and best
results in predicting multi-headed dependencies in a shared task of Chinese semantic dependency parsing.

In order to conduct a construction-specific nonlocal dependency recovery evaluation in Mandarin Chinese, a set of test sets is annotated for eight nonlocal constructions in Mandarin Chinese according to their annotations in the Penn Chinese Treebank. The coverage evaluation across different dependency formalisms shows that the Chinese GCG annotations have the best coverage of the nonlocal dependencies annotated in the test sets compared with other dependency formalisms such as Chinese Stanford dependencies or the Penn2Malt dependencies. The nonlocal dependency recovery evaluation across different parsers shows that a higher accuracy in general parsing evaluation such as evalb does not necessarily correlate to better performance in nonlocal dependency recovery. The construction-specific parsing evaluation shows that recovering nonlocal dependencies from frequent nonlocal constructions such as subject and object relative clauses, passive voice is easier than that from infrequent nonlocal constructions, such as topicalization or extraction from embedded clauses.
Dedicated to Yueshan and Yuechuan
Acknowledgments

My deepest gratitude goes to my advisors, Carl Pollard and William Schuler.

Carl was the first person from the department to contact me to let me know that I was admitted. In first email, he said that he was not able to contact me sooner because he was sidelined for a few days by a big ice storm that ‘knocked out the power and sent an icy tree limb through the roof and bedroom ceiling’. Carl went on counting many advantages and benefits of living in Columbus, but nothing can be more impressive than the snow storm, so I decided to go to Columbus. After I came, with a lot of ‘Chiquito’ and ‘donkey’ sentences, Carl brought me into this fascinating world of formal grammars. By constantly asking me to consider the grammaticality of Chinese sentences with various degrees of syntactic weirdness, Carl taught me how important to ground my research on solid linguistic facts. Carl has been the most knowledgeable linguist I have ever worked with, but he is always interested in knowing how I think. It has been such a pleasure and privilege to be his advisee. Carl is the best advisor I could ever ask for.

I have been always grateful that William Schuler agreed to serve as my advisor in my second year when I did not have any programming or engineering background but had this idea of working on Chinese parsing within a categorial grammar framework. He listened to me and then helped me, step by step, realize the goal, which ends up to be the thesis presented here. William is very resourceful in computational linguistics and many fields related to natural language processing. He is genuinely interested in linguistic research and never stops updating his knowledge with the latest developments in related fields. I have learned tremendously from working with him throughout the
years. William is the most patient and supportive advisor I have ever worked with. He gave the most detailed comments on all the drafts I ask him to review. He guides me through difficulties and encourages me to stay positive. Without him, I would have given up long time ago.

I want to thank Micha Elsner, who opened the door of machine learning for me. With the suspicion that how machine learns, I took his Computational Linguistics II class. It was a very challenging class for me, but Micha helped me survive it. For many times, I needed to go to his office to ask him to explain the lecture all over again. He never felt it was a burden and was always patient even he needed to explain very basic concepts. He also very generously shared with me his ‘Where is Wally’ data, based on which I completed my first machine learning project and published the results in the first paper of my Ph.D journey.

I am also grateful to Mike White for giving me a great chance to work on his NSF project, which offers me precious project experience. Mike trusted me to work on his new project when I did not even know how to use command lines. He patiently showed me how to get it started and always provided timely help when I get stuck. When serving on my second qualifying paper reading committee, Mike was observant in spotting errors that had eluded me and raised a lot of thought-provoking questions. His insightful comments directly shaped the work presented here.

My thanks also go to Zhiguo Xie and Marjorie Chan (Ma Laoshi) from the Department of East Asian Language and Literature. Despite of his busy schedule, Zhiguo agreed to serve on my candidacy and dissertation committee without hesitation. He stayed up late for my candidacy exam while he was traveling in China. He is a knowledgeable resource of Chinese linguistics and is always ready to discuss any Chinese linguistics questions I have. Ma Laoshi is the most warm-hearted and caring person I have ever met. She always makes sure to include me into various activities held in her department. She kindly invited me to several events of Chinese linguistics and introduced me to the colleagues from those events.
I want to thank Judith Tonhauser, Bob Levin and Yusuke Kubata for their interest in my research. Their invaluable comments and questions always push me to think deep. I want to thank Hope Dawson for giving me detailed feedbacks for my teaching and kindly writing references for my job applications. I want to thank Jim Harmon for helping me fix countless computer problems in a timely manner. I want to thank Julia Papke for her patient and prompt help along the way.

I am blessed to know so many wonderful friends at OSU. My friends, Abby Walker, Murat Yasavul, Stephanie Antetomaso, Shontael Elward, Evan Jaffe, David King, Nathan Rasmussen, Cory Shain, Chris Worth, Scott Marten, Qian Wang, to name too few, are always ready to help me and cheer me up. I thank Lifeng Jin for running several neural network models for me and, more importantly, treating me with most authentic Hunan and Sichuan cuisine I have ever had. Qinyang Yan and Yuhan Lin are always my listeners, rain or shine. At last, I want to reserve my special thanks to Marty van Schijndel and Liela Rotschy. Marty was the person who encouraged me to explore computational linguistics seriously. I am deeply grateful to Liela for her company and help in my most difficult times. They offered me too many sweet memories and life-long friendship.

I am enduringly grateful to my family for their unconditional love and support. From the moment I decided to give up my job back in China and start this journey of risks and uncertainties, my husband, Xiangjun, stood by my side. To support me, he resigned the job in China and came to the US to join us. He took over most of the housework so I can have more time learning. He taught me calculus and coding so I can catch up in class. He worries more about my progress than I do myself and constantly urges me to get things done on time. Without him, this thesis will not be finished until three years later. I am thankful to my two amazing kids, Yueshan and Yuechuan, whose heart-melting smiles and cozy cuddles always remind me what is really important. Finally, my forever gratitude goes to my parents. With their ‘blind’ confidence in me, they set me free to pursue what I want and be who I am.
Vita

2018 ..................................................... M.S. in Statistics
The Ohio State University

2003 ..................................................... M.A. in Linguistics
Zhejiang University

Publications

Research Publications


Fields of Study

Major Field: Linguistics
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Dedication</td>
<td>iv</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>v</td>
</tr>
<tr>
<td>Vita</td>
<td>viii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xi</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xiii</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2. Generalized Categorial Grammar</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Grammar formalism</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Generalized Categorial Grammar</td>
<td>7</td>
</tr>
<tr>
<td>2.2.1 Defining a GCG</td>
<td>8</td>
</tr>
<tr>
<td>2.2.2 Dependencies in GCG</td>
<td>8</td>
</tr>
<tr>
<td>2.3 Penn Treebank</td>
<td>14</td>
</tr>
<tr>
<td>3.1 Chinese Syntax in GCG</td>
<td>15</td>
</tr>
<tr>
<td>3.1.1 Argument composition</td>
<td>17</td>
</tr>
<tr>
<td>3.1.2 Modifier composition</td>
<td>18</td>
</tr>
<tr>
<td>3.1.3 Nominal and quantificational expressions</td>
<td>20</td>
</tr>
<tr>
<td>3.1.4 Topicalization</td>
<td>22</td>
</tr>
<tr>
<td>3.1.5 Focus construction</td>
<td>26</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>3.1.6</td>
<td>Relative clauses</td>
</tr>
<tr>
<td>3.1.7</td>
<td>Appositive clauses</td>
</tr>
<tr>
<td>3.1.8</td>
<td><em>Ba</em> construction</td>
</tr>
<tr>
<td>3.1.9</td>
<td><em>bei</em> constructions</td>
</tr>
<tr>
<td>3.1.10</td>
<td>Conjunction</td>
</tr>
<tr>
<td>3.2</td>
<td>Summary</td>
</tr>
<tr>
<td>4.</td>
<td>Syntactic Parsing Evaluations</td>
</tr>
<tr>
<td>4.1</td>
<td>Reannotation</td>
</tr>
<tr>
<td>4.2</td>
<td>Parsing Experiments</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Parsing performance with different training sets</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Parsing performance comparison with CCG</td>
</tr>
<tr>
<td>4.3</td>
<td>Construction Specific Parsing Performance Analysis</td>
</tr>
<tr>
<td>4.4</td>
<td>Summary</td>
</tr>
<tr>
<td>5.</td>
<td>Semantic Dependency Parsing Evaluations</td>
</tr>
<tr>
<td>5.1</td>
<td>Chinese Semantic Dependency Annotations</td>
</tr>
<tr>
<td>5.2</td>
<td>GCG-based semantic dependency parsing system</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Parsing training data</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Multinomial Dependency Label Classifier</td>
</tr>
<tr>
<td>5.3</td>
<td>Results</td>
</tr>
<tr>
<td>5.4</td>
<td>Error Analysis</td>
</tr>
<tr>
<td>5.5</td>
<td>Summary</td>
</tr>
<tr>
<td>6.</td>
<td>Nonlocal Dependency Parsing Evaluations</td>
</tr>
<tr>
<td>6.1</td>
<td>Empty categories in Penn Chinese Treebank</td>
</tr>
<tr>
<td>6.2</td>
<td>Nonlocal constructions</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Possible Nonlocal Constructions</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Annotated Nonlocal Construction</td>
</tr>
<tr>
<td>6.2.3</td>
<td>The Data</td>
</tr>
<tr>
<td>6.3</td>
<td>Experiments</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Comparison between dependency formalisms</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Comparison between parsers</td>
</tr>
<tr>
<td>6.4</td>
<td>Error analysis</td>
</tr>
<tr>
<td>6.5</td>
<td>Summary</td>
</tr>
<tr>
<td>7.</td>
<td>Conclusion</td>
</tr>
</tbody>
</table>
### List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Train/Dev/Test Split</td>
</tr>
<tr>
<td>4.2</td>
<td>Parsing results on the development set</td>
</tr>
<tr>
<td>4.3</td>
<td>Parsing results on the test set</td>
</tr>
<tr>
<td>4.4</td>
<td>Parsing results, error reduction ratios and significance testing results on the common test set of NoUnary+NoLab trees.</td>
</tr>
<tr>
<td>4.5</td>
<td>Parsing results on the development set where ba sentences are annotated with either CCG analysis (ccg.ba) or GCG analysis (gcg.ba)</td>
</tr>
<tr>
<td>5.1</td>
<td>Distribution of the corpora</td>
</tr>
<tr>
<td>5.2</td>
<td>Data format of SemEval 16, Task 9</td>
</tr>
<tr>
<td>5.3</td>
<td>Results of development set where LF is the F1 score of the labeled dependency, NLF is the F1 score of the non-local dependency, UF is the F1 score of the unlabeled dependency and NUF is the F1 score the non-local unlabeled dependency.</td>
</tr>
<tr>
<td>5.4</td>
<td>Results of the test set compared with the best system where GCG is the current system and TOP is the system with the best labeled F1 score.</td>
</tr>
<tr>
<td>6.1</td>
<td>Empty categories in the Penn Chinese Treebank</td>
</tr>
<tr>
<td>6.2</td>
<td>Test data size and frequency of the constructions.</td>
</tr>
<tr>
<td>6.3</td>
<td>Nonlocal dependency coverage rates of Stanford dependency (stf), Penn2Malt dependency (p2m) and GCG dependencies converted from gold Penn Chinese Treebank trees and the coverage rate from an automatic parser trained on GCG annotations of Mandarin Chinese (gcg-a).</td>
</tr>
</tbody>
</table>
6.4 Nonlocal dependency coverage rates of Berkeley latent-variable parser (bkl) and the Berkeley self-attentive neural network parser (bnp) trained on a small (-s) and a large (-l) training set. ................................................................. 106

6.5 Evalb F1 scores for the two parsers trained on the small training set ............................... 107
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Treebank annotation for 国有企业活力不足 ‘State-owned enterprises do not show much vitality’</td>
</tr>
<tr>
<td>3.2</td>
<td>GCG annotations for 国有企业活力不足 ‘State-owned enterprises do not show much vitality’</td>
</tr>
<tr>
<td>3.3</td>
<td>GCG derivation:“Maotaijiu, which is titled as the Chinese national liquor, has a long history” (a) and its associated dependencies (b)</td>
</tr>
<tr>
<td>3.4</td>
<td>Treebank annotations for : ‘Dupont established four agencies and three foreign-owned enterprises.’</td>
</tr>
<tr>
<td>3.5</td>
<td>GCG annotations for : ‘Dupont established four agencies and three foreign-owned enterprises.’</td>
</tr>
<tr>
<td>4.1</td>
<td>Annotating relative clause structure in the sentence “Chongming has a position that is superb”</td>
</tr>
<tr>
<td>4.2</td>
<td>Constructing common test set</td>
</tr>
<tr>
<td>4.3</td>
<td>Derivations of the sentence We will push the reform as a whole to depth.</td>
</tr>
<tr>
<td>5.1</td>
<td>An example of semantic dependency annotations in SemEval 2016, Task 9</td>
</tr>
<tr>
<td>5.2</td>
<td>Tree structures for 国际货币基金组织 ‘International Monetary Fund Organization’</td>
</tr>
<tr>
<td>5.3</td>
<td>Semantic dependencies for 杰里米喜欢做的事情 ‘things that Jimmy likes to do’</td>
</tr>
<tr>
<td>6.1</td>
<td>Treebank annotations for 美国对华贸易逆差被严重夸大 ‘The US trade deficits with China have been greatly exaggerated.’</td>
</tr>
</tbody>
</table>
6.2 Treebank annotations for 外商投资企业获得的人民币贷款 ‘The loan in Chinese RMB that foreign invested enterprises obtained’ ........................... 77
6.3 Treebank annotations for 投资额较大的台资企业 ‘The Taiwan enterprises whose investment amounts are comparatively large’ ........................... 77
6.4 Treebank annotations for 中国经济最为活跃的地区 ‘The areas where the economy is most active in China’ ........................... 78
6.5 Treebank annotations for 张三被大学录取 ‘Zhangsan was admitted by a university.’ 78
6.6 Treebank annotations for “我们会赢的”，张三说 ‘We will win.” said Zhangsan’ 79
6.7 Treebank annotations for 张三好像被打了一手 ‘Zhangsan seems to have been beaten’ . 79
6.8 Treebank annotations for 发展国民经济是头等大事 ‘Developing the national economy is an important thing of the top priority’ ........................... 80
6.9 Treebank annotations for ‘中国设法发展科技’ ‘China tries to develop science and technology’ ........................... 81
6.10 Treebank annotations for 张三请李四吃饭 ‘Zhangsan invites Lisi to have a dinner’ 82
6.11 Treebank annotations for 关门 ‘Close the door’ ........................... 83
6.12 Treebank annotations for 积极引进，精心种植名贵品种‘proactively introduce and carefully plant precious species’ ........................... 83
6.13 Treebank annotations for 把崇明建成港口 ‘build Chongming into a port’ 85
6.14 Treebank annotation for 连美国警署都从这里引进警犬 ‘Even American police stations use police dogs from here’ ........................... 86
6.15 Treebank annotation for 他连我都不认识 ‘He could not even recognize me’ . 86
6.16 Treebank annotation for 他作业做完了 ‘He finished his homework’ ........................... 87
6.17 Treebank annotation for 是谁拯救了中国人民？ ‘Who saved the Chinese people?’ 89
6.18 Treebank annotation for 大量出现的是以前不曾出现的新情况 ‘what has emerged a lot was new situations that have never emerged before’ .......................... 90

6.19 GCG analysis for 喜欢李四的是张三 ‘the one who likes Lisi is Zhangsan’ .................. 90

6.20 Stanford dependencies for 股市不断下跌投资人信心全失 ‘The stock market continues to fall and the investors completely lost their confidence’ .......................... 102

6.21 Stanford dependencies for 星期三预料是克林顿外交之行的高潮 ‘Wednesday is expected to be the climax of Clinton’s diplomatic visit’ .......................... 103

6.22 Predicted parses for 俩人被判处无期徒刑 ‘The two people was sentenced to life imprisonment’ .................. 109

6.23 Treebank annotations for 新的忧患 ‘new worries’ .......................... 110

6.24 Treebank annotations for 新的特色 ‘new features’ .......................... 111

6.25 A possible parse for noun + verb + de + noun + de + noun .......................... 112

6.26 A possible parse for noun + verb + de + noun + de + noun .......................... 112
Chapter 1: Introduction

Syntactic dependencies can be local or nonlocal. Local syntactic dependencies occur within the same clause and can be inferred from the predicate-argument structures. Nonlocal dependencies involve two positions whose correspondence cannot be directly inferred from the predicate-argument structure. In Mandarin Chinese, nonlocal dependencies can be found in syntactic constructions such as relative clause, topicalization, passive voice or focus construction. In the Penn Chinese Treebank (Xue et al., 2005), trace categories are annotated for some constituent to indicate the location where the constituent can be interpreted. The trace categories in the Penn Chinese Treebank include traces left by A movement and A-bar movement defined in phrase structure grammar (Chomsky, 1981).

Correctly resolving nonlocal dependencies is crucial in understanding the underlying predicate-argument structure of a sentence, which is useful for any downstream information extraction application where accurate understanding of the predicate-argument structure of a sentence is crucial. This is especially important in Chinese, since Chinese makes heavy use of nonlocal dependencies and since recovering them has been identified as a difficult problem (Kummerfeld et al., 2013).

This research explores resolving nonlocal dependencies in Chinese by parsing the language using Generalized Categorial Grammar (GCG) annotations (Bach, 1981; Nguyen et al., 2012). Categorial grammar annotations are attractive because they have a transparent syntactic-semantic interface and provide a natural account for filler-gap phenomena (Nguyen et al., 2012; Rimell
The GCG annotations, which are obtained from reannotating Penn Chinese Treebank (Xue et al., 2005), retain the information of trace categories in the Treebank annotations to make it possible for the parser trained on them to recover nonlocal dependencies.

The GCG framework here uses a slightly larger set of language-specific inference rules and a substantially smaller category set than strongly lexicalized grammars like Combinatory Categorial Grammar (Steedman, 2000, 2012, CCG), which makes it suffer less from the sparse data problem. Experimental results show a statistically significant gain in parsing accuracy from this moderately lexicalized grammar over parsing with CCG (Duan and Schuler, 2015; Nguyen et al., 2012).

We choose to resolve nonlocal dependencies using a Chinese GCG framework because syntactic dependencies can be inferred from the GCG derivations in a straightforward way. The gap operator -g in this framework records the location of the trace category annotated in the Treebank and allows all the nonlocal dependencies annotated in the original Treebank to be retained.

Other categorial grammar frameworks may also provide nonlocal dependency information. The only existing categorial grammar annotations for Chinese is Chinese CCGbank (Tse and Curran, 2010). However, the current release of the Chinese CCGbank does not automatically generate dependencies, which makes it difficult to use it in dependency evaluation.

Stanford dependencies for Mandarin Chinese (Chang, 2009; Chang et al., 2009) and the later proposed Universal Dependencies (de Marneffe et al., 2014; Nivre et al., 2016) are popular choices for the evaluation of Chinese dependency parsing. However, the conversion from Penn Chinese Treebank to Stanford dependencies loses the trace information annotated in Treebank. The universal dependency project represents an effort to facilitate multilingual research with one unified annotation schema for all languages. However, like Stanford dependencies, universal dependencies for Chinese do not include dependencies from or to trace categories annotated in the Penn Chinese Treebank. Also in order to build dependencies between only content words, universal dependencies retain
less information in constructions such as coordination and prepositional phrases, which sometime causes ambiguity in extracting dependencies.

This thesis presents the development of a system of resolving nonlocal dependencies through automatic parsers trained on Chinese GCG annotations. Following this introduction, the thesis proceeds as follows.

• Chapter 2 describes the categorial grammar framework and the dependencies it yields. It also briefly introduces the Penn Treebank annotations, from which the broad-coverage GCG annotations are obtained.

• Chapter 3 describes the details of the generalized categorial grammar formalism designed for Mandarin Chinese and the analyses of some language-specific constructions in Mandarin Chinese in this framework.

• Chapter 4 introduces the reannotation from the Penn Chinese Treebank into Chinese GCG annotations and presents the evaluation of a syntactic parser trained on the Chinese GCG annotations. The syntactic parsing evaluation compares the parsing performance of a latent-variable parser (Petrov and Klein, 2007) trained on Chinese GCG annotations with that of the same parser trained on Chinese CCG annotations. The parser trained on GCG annotations performs significantly better than the same parser trained on the CCG annotations on a common test set of unlabeled binary structures (Duan and Schuler, 2015).

• Chapter 5 presents an evaluation of a semantic dependency parsing system established on the Chinese GCG annotations. The system yields overall comparable results as the state-of-the-art system and superior results on the dependencies which involve multiple heads.
Chapter 6 provides the details of a nonlocal dependencies recovery experiment using syntactic parsers trained on Chinese GCG annotations. We compare the recovery rates of different dependency formalisms and nonlocal dependency recovery performances of different parsers.
Chapter 2: Generalized Categorial Grammar

2.1 Grammar formalism

Grammar formalisms are designed to produce latent syntactic structures of a language either to provide an interface for extracting meaning representations out of the language or to account for the legal surface strings of the language. These formal systems are evaluated on their capacity to describe different linguistic phenomena and their implementability or learnability. In this research, we are interested in how nonlocal constructions are accounted for in these formalisms and whether the nonlocal dependencies are easily learnable from the annotations of the grammar formalism.

Transformational grammar (Chomsky, 1965, 1981) proposes deep structure and surface structure as two different syntactic levels for languages and accounts for nonlocal constructions with movement which only occurs in the surface structure. Various movement theories are proposed to account for the discontinuities in specific constructions and locality theory is proposed to filter or block certain types of movement. Transformational grammar does not provide a unified mechanism for the discontinuities found in different constructions. In practice, none of the start-of-the-art syntactic parsers learns to propose the trace category left by the movement.

Another group of syntax theories accounts for nonlocal construction with feature passing. The interpretation of displaced constituents depends on the propagation of features along the unifications or derivations of the linguistic units. In head-driven phrase structure grammar (HPSG; Pollard and
Pollard and Sag show that the feature structure of a unification based grammar can account for the nonlocal constructions without resorting to any movement theories. In HPSG, the SLASH attribute is used to regulate the grammatical relation between the filler and the gap in a nonlocal dependency. SLASH is a nonlocal attribute which propagates from a child node to the parent node in the syntactic tree until it is unified with a sibling whose local value matches the slash value. The SLASH values represent gaps and the siblings with the matching local attributes are the fillers in the nonlocal dependencies. Unification based on value matching establishes a nonlocal dependency between the filler and the gap.

Categorial grammar is another family of natural language formalisms, where nonlocal constructions are accounted for without resorting to any theories of movement. Motivated by the principle of compositionality of Frege (1892), a categorial grammar formalism combines syntactic constituents into a larger constituent using inference rules. The two important components of a categorial grammar are the lexicon and the inference rules. Lexicons are sets of terminal symbols which are assigned categories or types. Inference rules are sets of rules defining how symbols of different categories combine and what semantic interpretations or dependency representations they can derive. Compared with generative grammars, the categories of terminal symbols in categorial grammars are much more informative, containing important information about how each symbol combines with others.

The simplest categorial grammar, which only contains two inference rules, forward and backward application, is called AB categorial grammar, after Ajdukiewicz (1935) and Bar-Hillel (1953). With only two inference rules, AB categorial grammar is not able to account for some linguistic phenomena such as nonlocal constructions in natural languages. Variants have increased the expressive power of AB categorial grammar by including more inference rules. (Steedman, 2000) augmented AB categorial grammar with composition and type-raising to account for nonlocal
constructions and various other linguistic phenomena. The resulting categorial grammar is called Combinatory Categorial Grammar (CCG; Steedman, 2000).

Introducing composition and type-raising inference rules into AB categorial grammar is not the only solution for accounting for nonlocal constructions in categorial grammar formalism. Linear Categorial Grammar (LCG; Pollard, 2013) is able to account for nonlocal constructions with its two inference rules, modus ponens and hypothetical proof, and a logical axiom for representing traces. In LCG, a trace category is introduced into the derivation by a logical axiom, which will propagate along the derivation until the hypothetical sign of the trace is discharged in an application of hypothetical proof, which will then be satisfied by the sign of the filler in an application of modus ponens to complete the long-distance association between the gap and filler.

2.2 Generalized Categorial Grammar

The grammar formalism we use in this research is Generalized Categorial Grammar (GCG; Bach, 1981). Similar to the idea of SLASH attributes from HPSG and hypothetical proof in LCG, GCG models the nonlocal constructions by introducing a gap category and a set of gap-related inference rules into the grammar. A gap category is introduced where the trace is and propagates in the derivation, and finally is discharged by the filler.

Like other categorial grammars, the GCG formalism we defined here contains a set of inference rules defining the combination of different categories and the interpretation of the combination. It also contains a lexicon with types or categories. However, different from CCG or LCG, GCG has a slightly larger set of language-specific inference rules. A larger set of language-specific inference rules makes GCG less lexicalized as CCG. For example, GCG has separate inference rules for argument composition and modification. This distinction makes the category of a modifier reusable even when the modifier occurs in different locations of a sentence. By contrast, in a strongly
lexicalized grammar like CCG, modifiers need to have different categories for each location that modifier possibly occupies in a sentence. For example, an adverbial phrase can occur at sentence initial position or at a preverbal position in Mandarin Chinese, with barely any difference in meaning. A strongly lexicalized categorial grammar needs to provide two different categories for this lexical item. Having a slightly larger set of inference rules helps to reduce the set of possible categories in a language, which can relieve the sparsity problem for automatic parsing.

2.2.1 Defining a GCG

A generalized categorial grammar (GCG; Bach, 1981; Nguyen et al., 2012) is a tuple \( \langle P, O, R, W, M \rangle \) consisting of a set \( P \) of primitive syntactic categories, a set \( O \) of type-constructing operators, a set \( R \) of inference rules, a set \( W \) of vocabulary items, and a mapping \( M \) from vocabulary items to their syntactic categories. A set of syntactic categories \( C \) is defined as: \( P \subset C; C \times O \times C \subset C; \) nothing else is in \( C \).

\( M \) is a function that maps a vocabulary item \( w \in W \) to an ordered pair whose first component is its syntactic category \( c \) and whose second component is its syntactic dependency representation \( g \), using the notation \( w \mapsto c : g \). The syntactic dependency representation \( g \) is expressed in dependency functions, labeled with dependency types or argument position numbers: \( f_0, f_1, f_2, \) etc. For example, the dependency representation of word 'cat' is \( \lambda x (f_0 x)=cat \), which contains a single ‘0’-labeled dependency to a constant ‘cat’.

2.2.2 Dependencies in GCG

In categorial grammars, the composition of syntax is usually coupled with the composition of semantics. This coupling gives categorial grammars a transparent syntax-semantics interface property (Steedman, 2000). The current GCG framework composes syntactic dependencies along with syntactic composition. This section discusses the relative merits and deficiencies of this
representation compared to other representations of syntactic or semantic relations, including Stanford/Universal Dependencies (de Marneffe et al., 2014), Meaning Text Theory (Mel’čuk, 1988), Minimal Recursion Semantics (Copestake et al., 2005), Abstract Meaning Representation (Banarescu et al., 2013), Hybrid Dependency Logic (White, 2006), and the cued-association semantics described by Schuler and Wheeler (2014).

First, the fact that the syntactic dependencies can be reliably and easily extracted from the GCG derivations makes them a natural choice for dependency representation. Since the gap categories are implemented in the derivations, the syntactic dependencies extracted from the derivations provide sufficient information for the nonlocal dependency parsing evaluation task.

Secondly, the syntactic dependencies extracted in this way can serve as a starting point to establish deep syntactic or semantic dependencies when combined with construction specific information or lexical information. The fact that a semantic dependency parsing system built upon the syntactic dependencies extracted from Chinese GCG derivations yields comparable overall performance to top systems and better accuracy for dependencies with multiple heads proves the utility of these syntactic dependencies (Duan et al., 2016).

The syntactic dependencies shown here can also be interpreted as a shorthand for distributed representations of sentence meanings compatible with cognitive computational neuroscientific models of episodic memory Schuler and Wheeler (2014).

Stanford dependencies (de Marneffe et al., 2006) and universal dependencies (de Marneffe et al., 2014), due to their convenient conversion from Penn Treebank, have been widely used as a standard dependency format in various parser evaluations. However, I did not use Stanford/Universal dependencies for nonlocal dependency parsing evaluation because Stanford/Universal dependencies, when converted from Penn Chinese Treebank, lose important trace information annotated in Treebank.
Rimell et al. (2009) have the same problem when evaluating nonlocal dependency parsing performance in English across different dependency formats. For example, the nonlocal dependency evaluation task requires more fine-grained classification of relative clauses while many dependency formats, including Stanford dependencies, adopt a generic ‘relative modifier’ label \texttt{rcmod}. Rimell et al. (2009) solved this problem by some heuristic post-processing. When \texttt{rcmod}(N1, V) is present in the dependency parsing output, indicating that the head noun \textit{N1} is modified by a relative clause with the main verb \textit{V}, they infer the relative clause is a subject relative clause if they also find a dependency \texttt{dobj}(V, N2) or an object relative clause if they find a dependency \texttt{nsubj}(V, N3). However, the recovery of this information is harder for Chinese because Chinese is a pro-drop language and Chinese has some language-specific nonlocal constructions such as topic relative clauses. These characteristics make it harder to infer the exact relative clause types unless the extraction sites are clearly marked. For example, for the relative clauses in (1a) and (1b), Stanford dependencies will have the same dependency \texttt{rcmod}(鱼 ‘fish’, 吃 ‘eat’) for both of them, while the GCG dependencies will have \texttt{2}(鱼 ‘fish’, 吃 ‘eat’) for (1a) and \texttt{1}(鱼 ‘fish’, 吃 ‘eat’) for (1b) to indicate ‘fish’ is the second argument of ‘eat’ in (1a) and the first argument of ‘eat’ in (1b).

(1) a. 张三点了一些爱吃的鱼
   Zhangsan order ASP some love eat \textit{de} fish
   ‘Zhangsan ordered some fish that he likes’

   b. 爱吃的鱼到处找吃的
   love eat \textit{de} fish everywhere look-for eat \textit{de}
   ‘Fish that love eating are looking for food everywhere’

(2) 胃口难以改变的人不适合旅行
   appetite hard-to-change \textit{de} people not suitable travel
   ‘People whose appetite is hard to change are not suitable for traveling’
The Stanford dependencies will contain \texttt{rcmod(人 ‘people’, 难以改变 ‘hard-to-change’)} for (2). With this dependency it is hard to decide whether the head noun ‘people’ is the object or the topic of the relative clause. The syntactic dependencies extracted from GCG derivations of the sentence will mark ‘people’ as the topic of the relative clause by the dependency \texttt{of-asso(胃口 ‘appetite’, 人 ‘people’)}.

Meaning Text Model (MTM; Mel’čuk, 1988) distinguishes seven levels of representations from meaning to physical forms of utterances. From deep to surface, there are semantic representations, deep syntactic representation, surface syntactic presentation, deep morphological representation, surface morphological representation, deep phonetic representation and surface phonetic representation. Correspondingly, for dependencies, Meaning Text Theory differentiates dependencies on all seven levels.

The syntactic dependency representation used for the GCG annotations here falls more into the category of surface syntactic representation than the deep syntactic representation according to the criteria in Mel’čuk (1988). In MTM, a node of deep syntactic structure is labeled with a generalized (deep) lexeme and deep syntactic relations are supposed to be language-independent. The dependency representation used in this research is language specific and uses actual lexemes in Chinese. Although numbered dependency types are used in the proposal, they are different from the argument relations, notated as 1, 2, ..., 6 in the deep syntactic dependency representation in Mel’čuk (1988). In MTM, the 1st, 2nd, ..., 6th argument of a functor each refers to a class of syntactic constructions serving the same semantic argument of a predicate. The numbered dependency types in this research do not have this semantic implication.

Abstract Meaning Representation (AMR; Banarescu et al., 2013) is proposed with the aim to create a simple readable semantic bank of English sentences paired with their whole-sentence, logical meanings. AMRs are rooted, labeled graphs which abstract away from syntactic idiosyncrasies.
Sentences such as *He described her as a genius*, *His description of her: genius*, and *She was a genius, according to his description* are all assigned with the same AMR. AMR makes extensive use of Propbank frame sets (Kingsbury and Palmer, 2002) to label the relations in semantic graphs. Approximately 100 relations and the inverses of them are employed in AMR graphs. The fact that AMR is agnostic about how to derive meanings from strings makes it hard to employ AMR as the semantic representation for any categorial grammar framework, in which meaning is built up compositionally.

Minimal Recursion Semantics (MRS; Copestake et al., 2005) is proposed to meet the demands for a semantic representation framework which is easy to decompose, relate and compare. The semantic representation also needs to be scope underspecified because quantifier scope cannot be obtained from compositional processing directly and it might not be needed for some application such as machine translation. However, the representation also needs to preserve sufficient information about scope so the possible scopal reading can be constructed when it is needed. In order to address these needs, MRS is created to have a flat semantic representation of lists of conjoined elementary predications (EP), with no EP embedded by another, where the order of EPs does not matter. Flattening the semantic representation is made possible by introducing *handles* for each EP to indicate the particular tree node it belongs to. Handles are introduced for quantifiers to match a scopal argument with an EP or a group of EPs. Underspecification of scope is achieved by using unresolved handles and specification of scope is achieved by introducing constraints (=q) which equate unresolved handles with certain handles. MRS can be naturally expressed in terms of feature-based grammar such as HPSG.

Hybrid Logic Dependency Semantics (HLDS; Baldridge and Kruijff, 2002; White, 2006), closely related to MRS, is a dependency-based semantic representation based on hybrid modal logic. Hybrid modal logic is an extension of standard modal logic with the introduction of *nominals*, a
basic formula which enables explicit reference to states at which a proposition holds. Like MRS, HLDS represents meaning in a flat conjunction of the heads and dependents to ease the checking of equivalence. The same scope underspecification and specification can also be achieved in HLDS by introducing $=_{q}$ as a modal relation between nominals.

Both MRS and HLDS are capable of describing various semantic phenomena, including nonlocal dependencies, coordination, and generalized quantifiers. These representations can also be built up compositionally using a unification-based framework such as Combinatory Categorial Grammar or Head-Driven Phrase Structure Grammar. However, these semantic representations are defined at the computational level Marr (1982), concerning what language processing tasks are and how to solve them with formal techniques allowed by a particular framework. The cued-association semantics Schuler and Wheeler (2014) is proposed at the algorithm level with the intention to model how these tasks are implemented by human language users. In cue-association semantics, dependencies are cued associations between mental states and those mental states are referential states that generalize over objects/stimuli that produce the same pattern of activation.

The syntactic dependencies extracted from the GCG derivations can be associated with the semantic dependencies in the cued-association semantics. The participants of some eventuality represented in semantic dependencies can be associated with the arguments in syntactic dependencies. For example, the third argument of $bei$ particle in passive voice of Mandarin Chinese can be modeled as the subject of the eventuality encoded by the second argument of the $bei$ particle. We can simply add a constraint to establish the semantic dependency between the subject and the eventuality introduced by the second argument. The mapping from syntactic dependencies to the semantic dependencies varies across different language-specific syntactic constructions.
2.3 Penn Treebank

Training a supervised statistical parser depends on the availability of a broad-coverage corpus. In this research, we use the Penn Chinese Treebank as our source of corpus. Penn Chinese Treebank 6 contains around 2,000 text files, 28,000 sentences and 780,000 words of news articles from Xinhua News Agency, Sinorama Magazine, and other news programs. This broad-coverage corpus is annotated with full syntactic trees according to the syntactic theory of transformational grammar (Chomsky, 1981) and created with a large amount of human annotation efforts. The Penn Treebank corpus has become the gold standard for syntactic annotations that various parsing models compete to predict in the field of computational linguistics.

Researchers who work with other syntactic formalisms, rather than transformational grammar, are fully aware of the difficulty of creating syntactically fully annotated wide-coveraging corpus of their own. Therefore, the Penn Treebank, with different sets of mapping rules, has been reannotated into other grammar formalisms including Head-driven Phrase Structure Grammar (Miyao et al., 2004), Lexical Functional Grammar (Cahill et al., 2004) and Combinatory Categorial Grammar (Hockenmaier, 2003; Hockenmaier and Steedman, 2007; Tse and Curran, 2010, 2012).
Chapter 3: Mandarin Chinese in a Generalized Categorial Grammar Framework

This chapter describes a generalized categorial grammar for Mandarin Chinese. Like other categorial grammars, this formalism has transparent predicate-argument dependencies, but is generalized to include language-specific type-constructing operators and inference rules so as to reuse the categories of syntactically indistinguishable signs occurring at different syntactic locations.

3.1 Chinese Syntax in GCG

The set of primitive syntactic categories for Mandarin Chinese, $P$, is a subset of all syntactic categories $C$, and contains the following categories, generally labeled with the part of speech of the head of the category:
V: verb-headed clause
   e.g. 猫 ‘cat’ 吃了 ‘ate’ 鱼 ‘fish’, ‘The cat ate the fish’

N: noun-headed phrase or clause
   e.g. 猫 ‘cat’ 吃了 ‘ate’ 鱼 ‘fish’, ‘The cat ate the fish’

D: de-clause
   e.g. 数学 ‘math’ 非常 ‘very hard’ 的 DE 想法 ‘ideas’, ‘the ideas that math is very hard’

C: cardinal number
   e.g. 三 ‘three’ 只 CL 猫 ‘cat’. ‘three cats’

Q: quantificational phrase
   e.g. 三 ‘three’ 只 CL 猫 ‘cat’, ‘three cats’

A: adjectival phrase or nominal modifier
   e.g. 小 ‘little’ 猫 ‘cat’, ‘little cats’

R: adverbial phrase or verbal modifier
   e.g. 他 ‘he’ 很快 ‘soon’ 走了 ‘walk away’, ‘He soon walked away’

B: verbal complement in ba construction
   e.g. 猫 ‘cat’ 把 BA 鱼 ‘fish’ 吃了 ‘ate’, ‘The cat ate the fish’

E: verbal complement in passive voice
   e.g. 鱼 ‘fish’ 被 BEI 猫 ‘cat’ 吃了 ‘ate’, ‘The cat ate the fish’

The set of type-constructing operators $O$ for Mandarin Chinese includes $-a$ and $-b$ operators for unsatisfied requirements of preceding or succeeding arguments, $-c$ and $-d$ operators for unsatisfied requirements of preceding or succeeding conjuncts, and a $-g$ operator for unsatisfied requirements of gap categories. Following Nguyen et al. (2012), directional operators such as forward and backward slashes (‘\’ and ‘/’) are not used because some operators, such as gap operators in tough constructions, are undirected.

A GCG category consists of a primitive category followed by one or more unsatisfied dependencies, each consisting of an operator followed by another category. For example, the syntactic category for a transitive verb is $V-aN-bN$, which has two unsatisfied dependencies, $-aN$ and $-bN$. The verb needs to take a noun phrase as an argument from its right and then another noun phrase as an argument from its left.
The set of inference rules \( R \) is described below.

### 3.1.1 Argument composition

The basic operation of most categorial grammars is argument composition. However, unlike most categorial grammars, the GCG described in this paper defines composition rules to explicitly encode dependencies between lexical items. Specifically, inference rules for argument composition are defined as follows, where \( c \in C, \ p \in P \) and each \( \varphi \in \{-a, -b\} \times C \):

\[
\begin{align*}
  c: g & \quad p \varphi_{1..n-1} \cdot ac: h \Rightarrow p \varphi_{1..n-1} \cdot \lambda x.g(f_n x) \land (h x) & (\text{Aa}) \\
  p \varphi_{1..n-1} \cdot bc: g & \quad c: h \Rightarrow p \varphi_{1..n-1} \cdot \lambda x.(g x) \land h(f_n x) & (\text{Ab})
\end{align*}
\]

The first composition rule Aa stipulates that when a predicate \( h \) of category \( p \varphi_{1..n-1} \cdot ac \) takes a preceding argument \( g \) of category \( c \) as its \( n \)-th argument, the syntactic dependency that \( g \) is \( h \)'s \( n \)-th argument is added. The second composition rule Ab is an argument composition rule taking a succeeding argument.

(3) 张三 喜欢 语言学

Zhangsan like linguistics

‘Zhangsan likes linguistics’

\[
\begin{align*}
  \lambda x_1.(f_0 x_1) = \text{zhangsan} & \quad \lambda x_2.(f_0 x_2) = \text{like} & \quad \lambda x_3.(f_0 x_3) = \text{linguistics} \\
  \lambda x_2.(f_0 x_2) = \text{like} \land (f_0(f_2 x_2)) = \text{linguistics} & \quad \lambda x_2.(f_0 x_2) = \text{like} \land h(f_n x) & \quad (\text{Ab}) \\
  \lambda x_2.(f_0 x_2) = \text{like} \land f_0(f_2 x_2) = \text{linguistics} & \quad \lambda x_2.(f_0 x_2) = \text{like} \land (f_0(f_1 x_2)) = \text{zhangsan} & \quad (\text{Aa})
\end{align*}
\]

The derivation of the sentence 张三 喜欢 语言学 ‘Zhangsan likes linguistics’ first combines 喜欢 ‘like’ and 语言学 ‘linguistics’ by the inference rule Ab and adds 语言学 ‘linguistics’ as the second argument of the verb 喜欢 ‘like’. It then combines 喜欢 语言学 ‘like linguistics’ with 张三
‘Zhangsan’ by the inference rule Aa and adds 张三 ‘Zhangsan’ as the first argument of the verb 喜欢 ‘like’. The resulting syntactic dependencies $\lambda x_2.(f_0 x_2) = \text{like} \land f_0(f_2 x_2) = \text{linguistics} \land f_0(f_1 x_2) = \text{zhangsan}$ shows that the event $x_2$ has a ‘0’-labeled dependency to a constant 喜欢 ‘like’, the first argument of $x_2$ has a ‘0’-labeled dependency to a constant 张三 ‘Zhangsan’ and the second argument of $x_2$ has a ‘0’-labeled dependency to a constant 语言学 ‘linguistics’.

3.1.2 Modifier composition

Inference rules for modifier composition apply preceding or succeeding modifiers of category $p-bd$ to modificands of category $c$, where $p \in \{A, R\}$, $d \in \{N, V\}$:

\[
\begin{align*}
\text{(Ma)} \\
& p-bd: g \ c:h \Rightarrow c: \lambda x \exists y (g(y) \land (h x) \land (f_1 y) = x) \\
\text{(Mb)} \\
& c: g \ p-bd: h \Rightarrow c: \lambda x \exists y (g(x) \land (h y) \land (f_1 y) = x)
\end{align*}
\]

The modifier composition rules Ma and Mb establish a ‘1’-labeled dependency from the modifier to the modificand, so the modificand is the first argument of the property represented by the modifier.
For example, for phrases such as 红苹果 ‘red apples’, 苹果 ‘apples’ is the first argument of the modifier 红 ‘red’. In this way, the phrase 红苹果 ‘red apples’ and the sentence 苹果红 ‘apples are red’ yield the identical dependency representations. The two modifier composition rules overgenerate, but the actual usage of the rules are constrained by the probability learned from the training data.

We illustrate the modifier composition rules with (4).

(4) 张三 很快 走 了
\[
\text{Zhangsan soon leave ASP}
\]
‘Zhangsan soon left.’
'soon' is the verb.

We can see from (4) and (5) that the adverb 很快 for CCG derivations of (4) and (5), we need to have two different categories for an adverbial modifier when it occurs in different syntactic positions. For example, in (4), 很快 ‘soon’ is a preverbal adverbial modifier, having the category R-bV. We can reuse the category R-bV when the modifier occurs at the sentence initial position as shown in (5).

(5) 很快 张三 走 了
soon Zhangsan leave ASP
‘Soon Zhangsan left.’

In the resulting dependency representation of (4), the first argument of the adverbial modifier 很快 ‘soon’ is the verb 走 ‘leave’.

Separating modifier composition rules from argument composition in GCG makes it possible to reuse modifier categories across different contexts. For example, in (4), 很快 ‘soon’ is a preverbal adverbial modifier, having the category R-bV. We can reuse the category R-bV when the modifier occurs at the sentence initial position as shown in (5).

We can see from (4) and (5) that the adverb 很快 ‘soon’ has the same syntactic category and the two sentences have the same syntactic dependency as we expected.

In contrast, in a strongly lexicalized grammar formalism such as CCG, we need to have different categories for an adverbial modifier when it occurs in different syntactic positions. For example, in CCG derivations of (4) and (5), we need to have two different categories, (S\NP)/(S\NP) and S/S for 很快 ‘soon’, even though these two occurrences are semantically indistinguishable.
For some commonly used Chinese prepositions and particles, the proliferation of the categories of the same word can be very severe in a strongly lexicalized grammar formalism. For example, Chinese CCGbank has 91 different categories for the character 在 ‘in/at’, because the prepositional phrase headed by 在 ‘in/at’ can modify constituents of various syntactic categories. In contrast, Chinese GCG annotations only have 9 different categories for 在 ‘in/at’.

Another example is the category of the aspect particle 了 in Chinese, which can either occur immediately after a verb or after the whole verb phrase to indicate that an action has completed. Although generalized backward crossed composition (Steedman, 2000) helps aspect particles in Chinese usually retain their canonical category \((S\setminus NP)\setminus(S\setminus NP)\), there are still 29 different categories for 了 in Chinese CCGbank (Tse and Curran, 2010), and most of them are semantically indistinguishable. In contrast, Chinese GCG annotations only have 1 category for 了 as an aspect marker.

### 3.1.3 Nominal and quantificational expressions

Mandarin Chinese does not have determiners such as ‘the’ or ‘a’ in English, so there is no empirical motivation to distinguish NP and N categories. However, *classifiers* or *measure words*, glossed as ‘M’ in (6), are obligatory when a noun is quantified by a number.

(6) 三 个 人

three classifier people

‘three people’

\[
\begin{align*}
\lambda x_1.(f_0 x_1) &= \text{three} \\
\lambda x_2.(f_0 x_2) &= \text{CL} \\
\lambda x_3.(f_0 x_3) &= \text{people} \\
\end{align*}
\]
We propose a separate category Q for quantificational expressions because they can be predicative, as in (7), which makes them different from common nouns. A zero-head rule Z, where \( c, d, e \in C \), converts the Q category to V-aN to make the quantificational expression predicative.

\[
e: g \Rightarrow c\text{-ad}: \lambda_x (f_0 x) = \text{pred} \land g(f_2 x)
\]

(7) 张三 三 岁 了
Zhangsan three years ASP
‘Zhangsan is three.’

Classifiers like 年 ‘years,’ 岁 ‘years-old’ and 天 ‘day’ already contain the nominal information, so they do not require nominal arguments like other classifiers. Classifiers of this type have a different category ‘Q-aC’ to reflect this combinational difference. By doing so, the numbers receive the same category C in both 三天 ‘three days,’ and 三个人 ‘three people.’ However, in both Chinese Treebank and CCGbank, the category ‘M’ is used for both types of classifiers, which results in numbers like 三 ‘three’ having the category QP/M in ‘three days’ and the category (NP/NP)/M in ‘three people’ in Chinese CCGbank. This is not desirable because it expends training examples on an artificial distinction between the numbers 三 ‘three’ in each of these expressions, which are semantically the same.
3.1.4 Topicalization

There are two types of topicalization in Mandarin Chinese. In the first type, there is a displaced constituent occurring at the sentence initial position. This type of topicalization is similar to that of English, in which the object is usually moved to the sentence initial position and a gap is left behind, as shown in (8).

Inference rules for gap composition are:

\[ p\varphi_{1..n-1}oc: g \Rightarrow p\varphi_{1..n-1}-gc: \lambda_{v_1}(g \cdot x) \land (f_n \cdot x) = v \]  
\[ c: g \Rightarrow c-gd: \lambda_{v_1}(g \cdot x) \land (f_1 \cdot v) = x \]  
\[ N: g \Rightarrow N-gN: \lambda_{v_1}(g \cdot x) \land \exists_e (\text{deasso} e \cdot x \cdot v) \]  

where \( p \in P \), \( o \in \{-a, -b\} \), \( c \in C \), \( d \in \{A-bN, R-bV\} \) and \( \varphi \in \{-a, -b\} \times C \). Rule Ga hypothesizes a gap as a preceding or succeeding argument, rule Gb hypothesizes an adjectival or adverbial modifier gap. Rule Ga and Gb are very similar to the slash introduction lexical rules in HPSG (Pollard, 1994) for argument and modifier ‘traces’. Rule Gc hypothesizes a gap which is associated with the subject in another type of topicalization in Chinese.

Non-local arguments, each consisting of a non-local operator and argument category \( \psi \in \{-g\} \times C \), are then propagated to consequents from all possible combinations of antecedents. For \( d: g \ e: h \Rightarrow c: (f \cdot g \cdot h) \in \{\text{Aa–b, Ma–b}\} \):

\[ d\psi_{1..m}: g \ e\psi_{m+1..n}: h \Rightarrow c\psi_{1..n}: \lambda_{v_1..n} f (g \cdot v_{1..m}) (h \cdot v_{m+1..n}) \]  

Rules Ac–d and Mc–d stipulate non-local propagation through argument and modifier composition.

Inference rules for filler attachment apply gapped clauses to topicalized phrases as fillers, which is similar to the topicalization rule of Gazdar (1981). For \( c \in C \), and \( p \in P \):

\[ p: g \ c-gp: h \Rightarrow c: \lambda_x \exists_y (g \cdot y) \land (h \cdot y \cdot x) \]  

( Fa)
The derivation of (8) is shown as follows, where the verb 吃 and the aspect particle 了 are separate tokens, shown together here to simplify the derivation. We apply the same simplification to 好 ‘good’ in (9).

(8) 饭, 我吃了
rice, I ate ASP
‘The rice, I ate.’

\[
\begin{align*}
\lambda x_1.f_0(x_1) &= \text{rice} \\
\lambda x_2.f_0(x_2) &= \text{I} \\
\lambda x_3.\exists y.(f_0 x_3) &= \text{eat} \land (f_0 y) = ASP \land (f_1 y) = x_3 \\
&\xrightarrow{\text{Ga}} \text{V-aN-bN:} \\
\lambda x_2.f_0(x_2) &= \text{I} \\
\lambda x_3.\exists y.(f_0 x_3) &= \text{eat} \land (f_0 y) = ASP \land (f_1 y) = x_3 \land (f_2 x_3) = v \\
&\xrightarrow{\text{V-aN-gN:}} \text{V-gN:} \\
\lambda x_3.\exists y.(f_0 x_3) &= \text{eat} \land (f_0 y) = ASP \land (f_1 y) = x_3 \land (f_2 x_3) = v \land (f_1 x_3) = I \\
&\xrightarrow{\text{V:}} \text{V:} \\
\lambda x_1.f_0(x_1) &= \text{rice} \\
\lambda x_2.f_0(x_2) &= \text{I} \\
\lambda x_3.\exists y.(f_0 x_3) &= \text{eat} \land (f_0 y) = ASP \land (f_1 y) = x_3 \land (f_2 x_3) = z \land (f_1 x_3) = I \\
&\xrightarrow{\text{Fa}} \text{V:} \\
\lambda x_3.\exists y.(f_0 x_3) &= \text{eat} \land (f_0 y) = ASP \land (f_1 y) = x_3 \land (f_2 x_3) = z \land (f_1 x_3) = I
\end{align*}
\]

In (8), a gap constituent is introduced by Ga when the second argument of the transitive verb 吃 ‘eat’ does not occur at its canonical post-verbal position. The gap propagates in the derivation until it is filled by the noun phrase 饭 ‘rice’. The analysis in (8) retains the dependency between ‘rice’ and ‘eat’ as expected.

The second type of topicalization occurs much more frequently in the Penn Chinese Treebank, in which a topic, usually a noun phrase, occurs at the sentence initial position to serve as the object of which the following sentence is ‘about’. Usually the subject of the sentence is ‘associated’ to the topic in some relationship.

(9) 他，胃口 很 好
he，appetite very good
‘Of him, the appetite is good.’
Usually, the referent of the subject in the non-movement topicalization needs to be further specified by the topic. For example, in (9), the subject 胃口 ‘appetite’ needs to be further specified by the topic 他 ‘he’ so as that the reader can know whose appetite is relevant here. Although topics are seen to be associated with other constituents of the sentence, especially in colloquial expressions, only associations with subjects are observed in the Treebank data. Therefore in our analysis of this type of topicalization, the subject undergoes a unary type conversion from N to N-gN to introduce a gap, which is later discharged by the topic to capture the ‘association’ relation between the subject and the topic.

In Chinese CCGbank, the topic in non-movement topicalization as in (9) is analyzed as a sentential modifier with the category S/S. In this analysis, the topic 他 ‘he’ is a sentential modifier for the sentence 胃口 很好 ‘appetite is good.’ This analysis conflates sentential adverbial modifiers such as ‘today’ with topics such as ‘he’ in (9), yielding an incorrect modifying relation between the topic and the main verb in the sentence. In (9), 他 ‘he’ is not semantically related to the predicate 很好 ‘good’ in any way.

In spite of its absence in the Penn Chinese Treebank corpus, some topicalization in Chinese does not see the close association between the topic and the subject, as shown in (10a).

(10) a. 水果，我 最 爱 吃 苹果
 fruit, I most like eat apples
 ‘As for fruit, I like to eat apples the most.’
Figure 3.1: Treebank annotation for 国有企业活力不足 ‘State-owned enterprises do not show much vitality’

Figure 3.2: GCG annotations for 国有企业活力不足 ‘State-owned enterprises do not show much vitality’

b. *我 最 爱 吃 水果 的 苹果
   I most like eat fruit de apple
   Not translatable

(11) a. 国有 企业  活力 不 足
   state-owned enterprises vitality not enough
   ‘State-owned enterprises do not have enough vitality’

b. 活力 不 足 的 国有 企业
   vitality not enough de state-owned enterprises
   ‘state-owned enterprises that do not have enough vitality’
In all topicalized sentences we manually checked in the Penn Chinese Treebank, the topic is associated with the subject of the sentence in a relation which usually can be expressed by the preposition of in English. In (11a), the subject 活力 ‘vitality’ is the vitality of the topic state-owned enterprises. Therefore, in GCG annotations of this type of topicalized sentences, the subject goes through a type change to introduce a gap which is filled by the topic, to capture the fact the subject needs further specification from the topic. The derivation of (11a) is shown in Figure 3.2.

However, this analysis dose not seem to be reasonable for (10a). The subject 我 ‘I’ does not need to be further specified by the topic 水果 ‘fruits’ here. It is possible that the topicalization in (10a) belongs to a different type of topicalization from (11a). For example, we can relativize the topic in (11a) to form a relative clause as shown in (11b), but the relative clause formed from (10a) sounds infelicitous, as shown in (10b). It seems the topic in (11a) is semantically more indispensable, while the topic in (10a) acts more like an adverbial modifier.

### 3.1.5 Focus construction

The inference rules proposed for the gap operations can also be used to analyze the focus construction in Mandarin Chinese. Focus construction, or sometimes called even-focus construction in Mandarin Chinese usually involves two morphemes 连 ‘even’ and 都 ‘all’ with 连 ‘even’ occurring before the focus, 都 ‘all’ occurring in a preverbal position and the 连 ‘even’ focalized noun phrase occurring before the verb. (13) is a typical even-focus construction where the focus 书包 ‘book-bag’ occurs in a preverbal position and 都 ‘all’ occurs before the verb.

(12) 张三 没 有 书包
Zhangsan not have book-bag
‘Zhangsan does not have a book-bag’

(13) 张三 连 书包 都 没 有
Zhangsan even book-bag all not have
‘Zhangsan does not even have a book-bag.’
In our analysis of the *even*-focus construction, we introduce a gap at the object position, which is filled by the focalized 连 phrase occurring before the verb. Following the Penn Chinese Treebank’s analysis, we treat 都 ‘all’ and 连 ‘even’ adverbial modifiers modifying the verb phrase and the focalized noun phrase respectively.

(a)

没有
not-have
V-aN-bN:
\[ \lambda x_5. (f_0. x_5) = \text{not-have} \]

R-bV:
\[ \lambda x_4. (f_0. x_4) = \text{all} \]

V-aN-gN:
\[ \lambda v x_5. (f_0. x_5) = \text{not-have} \land (f_2. x_5) = v \]

V-aN-gN:
\[ \lambda v x_5 \exists y. (f_0. x_5) = \text{not-have} \land (f_2. x_5) = v \land (f_0. z) = \text{all} \land (f_0. (f_1. z)) = x_5 \]

(b)

连
even
R-bV:
\[ \lambda x_2. (f_0. x_2) = \text{even} \]

N:
\[ \lambda x_3. (f_0. x_3) = \text{book-bag} \]

Ma

张三
Zhangsan
N:
\[ \lambda x_1. (f_0. x_1) = \text{Zhangsan} \]

V:
\[ \lambda x_5 \exists y \exists z. (f_0. v) = \text{book-bag} \land (f_0. z) = \text{even} \land (f_0. (f_1. z)) = v \land (f_0. x_5) = \text{not-have} \land (f_2. x_5) = v \land (f_0. z) = \text{all} \land (f_0. (f_1. z)) = x_5 \land (f_0. (f_1. x_5)) = \text{Zhangsan} \]

3.1.6 Relative clauses

In Mandarin relative clauses, the particle 的 ‘de’ takes a preceding clause containing a gap to form a relative clause modifying a succeeding noun. The modified noun is the filler of the gap.
in the relative clause. The inference rules for relative clauses apply the gapped \textit{de}-clause to the modificand as a filler.

For \( c \in C \):

\[
\text{D}-\text{gc}: \text{g N}: h \Rightarrow \text{N}: \lambda x. (h x) \land \exists y. (g x y)
\]

(\text{R})

A GCG analysis of a relative clause with an object gap is shown in (14).

(14) 猫 吃 的鱼
cat eat \textit{de} fish
‘fish that cats eat’

As in English, it is also possible to relativize a temporal or spatial noun phrase as an adverbial extraction to form a relative clause with the meaning ‘the place we met’ or ‘the time I saw you.’

The inference rule \( \text{R} \) works in the same way for these types of relative clauses, as shown in (15).

(15) 我们上班的时候
we work \textit{de} time
‘time when we work’
Our analysis of topicalization in (9) makes it easy to account for a relative clause which relativizes a topic. In (9) for example, relativizing the topic 他 ‘he’ yields a nominal phrase containing a non-restrictive relative clause 胃口很好的他, ‘he whose appetite is good.’ A GCG analysis of this nominal phrase is shown in (16).

(16) 胃口 好的 他
appetite good de he
‘he, who has a good appetite’
The analysis of topic relative clauses in Chinese CCGbank does not assume any gap or trace constituent as shown in (17). Therefore the topic 他 ‘he’ is not related to any constituent within the relative clause.

(17) CCG derivation of ‘he, who has a good appetite’

### 3.1.7 Appositive clauses

Appositive clauses in Mandarin Chinese are formed with the same 的 ‘de’ particle used in relative clauses. However, unlike relative clauses, appositive clauses do not involve any gap constituent. In this GCG analysis of appositive clauses, 的 ‘de’ receives the same category as it does in relative clauses. But the noun which takes an appositive clause as its complement has the category
N-aD to take a preceding de-clause to further specify the content of the noun. An appositive clause in this grammar is shown in (46).

(18) 高科技 高不可攀 的想法  

high-tech cannot-be-reached de idea  

‘the idea that high tech cannot be reached’

\[
\begin{align*}
N: & \quad \lambda x_1, (f_0, x_1) = \text{high-tech} \\
V-aN: & \quad \lambda x_2, (f_0, x_2) = \text{cannot-be-reached} \\
V: & \quad \lambda x_2, (f_0, x_2) = \text{cannot-be-reached} \land f_0(f_1, x_2) = \text{high-tech} \\
D-aV: & \quad \lambda x_3, (f_0, x_3) = \text{DE} \\
D: & \quad \lambda x_3, (f_0, x_3) = \text{DE} \land f_0(f_1, x_3) = \text{cannot-be-reached} \land f_0(f_1(f_1, x_3)) = \text{high-tech} \\
N-aD: & \quad \lambda x_4, (f_0, x_4) = \text{idea} \land f_0(f_1, x_4) = \text{DE} \land f_0(f_1(f_1, x_4)) = \text{cannot-be-reached} \land f_0(f_1(f_1(f_1, x_4))) = \text{high-tech}
\end{align*}
\]

In Chinese CCGbank appositive clauses have the same analysis as the topic relative clause.

(19) CCG derivation of ‘the idea that high-tech cannot be reached’

As shown in the examples above, relative clauses with a topic gap, adverbial gap and appositive clauses receive different analyses in our GCG formalism, while they are analyzed in the same way in the Chinese CCGbank.

3.1.8 *Ba* construction

*Ba* constructions in Mandarin Chinese require the affected patients of certain verbs to occur before the verb, instead of after the verb. For example, 鱼 ‘fish’ in (20) is the object of 吃 ‘eat’ and it occurs before the verb ‘eat’ in the *Ba* construction. In the Penn Treebank, 把 *ba* takes a
clause as argument. Therefore, 鱼吃了 ‘fish ate’ in (20) is analyzed as a clausal complement of ba. This analysis makes ‘fish’ the subject of the verb ‘eat,’ instead of the object. Consequently, Stanford dependencies extracted from Treebank annotations of this sentence have dependencies ‘nsubj (吃‘eat,’ 猫‘cat’),’ and ‘nsubj (吃‘eat,’ 鱼‘fish’),’ which is not correct.

In our analysis, we propose that the particle ba takes a ba-verb as its complement. Ba-verbs can be derived from transitive verbs with the type conversion rule given below.

c: g \Rightarrow d: g \quad (T)

The T rule is proposed to handle the situation where we need to change the syntactic category of a constituent. For example, the category of a sentence needs to be changed to be a noun phrase in the case of nominalization of a sentence. This rule is used with the constraints that the function g is preserved, no extra argument can be added to the dependency representation, and its usage is constrained by parsing probabilities for particular categories. Following Featherston (2005) and Crocker and Keller (2005), the model described here assumes that grammaticality judgments are gradient and determined by probabilities of compositional inferences occurring in the experience of...
a particular language user. Using the type conversion rule $T$, we change a transitive verb $V-aN-bN$ to $B-aN-aN$ to capture the fact that the verb that occurs within a $ba$ construction takes a preceding second argument.

The particle 把 $ba$ is assigned the category $V-aN-b(B-aN)$, with coindexation between the referent of its first argument ($f_1 \, x$) and the referent of the first argument of its second argument ($f_1 \,(f_2 \, x)$). Take (20) as an example, the lexicon entry of $ba$ establishes the coindexation between the referent of the first argument of the $ba$, i.e., $cat$, and reference of the first argument of $ate$, which means $cat$ is also the first argument of $ate$.

$$\text{把} \; \mapsto \; V-aN-b(B-aN): \lambda x. (f_0 \, x) = ba \land (f_1 \, x) = (f_1 \,(f_2 \, x)) \quad (ba)$$

Usually, the affected patient is the direct object of a transitive verb, as shown in (20), but there are cases where some verbs can only occur in $ba$ constructions. These types of verbs are $ba$-verbs with the category $B-aN-aN-bN$ and their category does not need to be converted from transitive verbs. Many resultative verbs ($VRD$, in treebank annotation) have this category. An example is given in (21).

(21) 我们 把 崇明 建成 港口
    we  $ba$ Chongming build-into port
    ‘We built Chongming into a port.’

(a).

(b)
In the analysis in (21), 我们 ‘we’ is the first argument of 成建 ‘build-into’; 崇明 ‘Chongming’ is the second argument of 成建 ‘build-into’ and 港口 ‘port’ is the third argument of 成建 ‘build-into’.

We have a different analysis of the ba-construction from the Chinese CCGBank, because we want to obtain the dependencies described as above for the sentences like (21). In Chinese CCGBank, the ba particle occurring in the ba-construction which contains a transitive verb has the category (S\NP_y)/(S\NP_y/NP_x)/NP_x, as shown in (22)\(^1\). Indices are used to derive the expected dependencies: the object of the transitive verb coindexes with the complement of ba and the subject of the transitive verb coindexes with the subject of ba. However, in the situation where a non-transitive verb occurs in the ba construction, as shown in (23), which is a simplified sentence from Chinese CCGBank, it remains unclear to us how the coindexing scheme would look. The derivation in (23) seems to suggest that 港口 ‘port’ is the second argument of the verb 成建 ‘build-into’ while 崇明 ‘Chongming’ does not hold any dependency with the verb ‘build-into’.

(22) CCG derivation of ‘the cat ate the fish’

\(^1\)Since coindices are not included within the released Chinese CCGBank, we infer this coindexing scheme from the examples provided in Tse and Curran (2010)
3.1.9  bei constructions

Mandarin Chinese uses the particle 被 bei to construct passive sentences. In bei constructions, the patient argument of a verb, usually the second argument of a transitive verb or a ba-verb, is moved to the subject position of the clause.

We propose the particle 被 bei takes a bei-verb as its complement. Bei-verbs, which are of the category E-aN-gN, are derived from E-aN-bN by introducing a gap by rule Ga. E-aN-bN is derived by the type conversion rule T from V-aN-bN or B-aN-aN, transitive verbs or ba-verbs. Here is the lexical entry we propose for the bei particle.

\[
\text{被 ‘bei’ } \mapsto V-aN-b(E-aN-gN)-bN : \lambda_x (f_0.x) = \text{bei} \land (f_3.x) = (f_1(f_2.x)) \quad (3.1)
\]

The lexical entry of 被 bei stipulates that the first argument of bei is the subject of its second argument, the VP complement, E-aN-gN. Since the agent in the passive voice construction is optional (as it is in passive voice in English), the category of the bei particle can have a type change from V-aN-b(E-aN-gN)-bN to V-aN-b(E-aN-gN).
The inference rule (Fc) is proposed for the composition of gap dependencies contained within succeeding arguments, where \( p \in P \), \( \varphi \in [-a, -b] \times C \), and \( \psi \in [-g] \times C \).

\[
p\varphi_{1 \ldots n-1} \cdot b(d\psi) : g \ d\psi : h \Rightarrow p\varphi : \lambda_x h(f_1 x)(f_n x) \land g x
\]  

(Fc)

Using rule Fc, the first argument of the bei particle becomes the filler of the gap in the bei verb. An example bei-construction which contains a transitive verb is shown in (24).

(24) 鱼 被 猫 吃了
fish bei cat ate
‘The fish was eaten by the cat.’

The Penn Treebank uses the category ‘LB’ for the bei particle where the optional agent argument occurs, and ‘SB’ for the bei particle where it is elided. Tse and Curran (2010) follow the Treebank annotations, proposing two different categories for the bei particle. For example the CCG category for ‘LB’ is \((S \backslash NP_y) / (S \backslash NP_x / NP_y) / NP_x\), in which a coindexation scheme is used to ensure that the subject of bei is coindexed with the object of its verbal complement. The ba particle, with the category \((S \backslash NP_y) / (S \backslash NP_y / NP_x) / NP_x\), is different from bei only in the coindexing scheme.

(25) CCG derivation of ‘The cat ate the fish.’
However, if the passivized verb is not a typical transitive verb as in (25), it is hard to infer the coindexing scheme. Take the passive voice of (21) as an example.

(26) 崇明 被 我们 建成了 港口

Chongming bei we build-into port

‘Chongming was build into a port by us.’

Example (26) shows the analysis of the passivized sentence of (21) in Chinese CCGBank. We can see that 港口 ‘port’ is the second argument of 建成 ‘build-into’ and 我们 ‘we’ is the first argument of 建成 ‘build-into’. However, 崇明 ‘Chongming’ does not hold any dependency with the verb, which we think is not desirable. Example (27) shows the GCG derivation of the same sentence and we can see that the dependency between 崇明 ‘Chongming’ and 建成 ‘build-into’ is established as we expect.

(27) 崇明 被 我们 建成了 港口

Chongming bei we build-into port

‘Chongming was build into a port by us.’
dependencies of p categories and what kind of dependencies we can extract from GCG derivations. Figure 3.3 shows a GCG derivation of a sentence from the Chinese Treebank. We use this sentence to illustrate how topicalization, passive voice, and relative clauses are analyzed in the GCG framework and what kind of dependencies we can extract from GCG derivations.

### 3.1.10 Conjunction

Inference rules for conjunctions use distinct -c and -d operators. The coordination of like categories $p\varphi_{1..n}\psi_{1..m}$, which have n unsatisfied arguments $\varphi_{1..n} \in \{-a,-b\} \times C$ and $m \geq 0$ non-local dependencies $\psi_{1..m} \in \{-g\} \times C$ and $p \in P$, is stipulated in the following inference rules:

\[ p\varphi_{1..n}\psi_{1..m} \vdash g \quad p\varphi_{1..n}\psi_{1..m} \cdot c(p\varphi_{1..n}\psi_{1..m}) \vdash h \]

\[ \Rightarrow p\varphi_{1..n}\psi_{1..m} \vdash \lambda v_{1..m} (g v_{1..m} (f_{n+1} x)) \land (h v_{1..m} x) \quad (Ca) \]

\[ p\varphi_{1..n}\psi_{1..m} \cdot d(p\varphi_{1..n}\psi_{1..m}) \vdash g \quad p\varphi_{1..n}\psi_{1..m} \vdash h \]

\[ \Rightarrow p\varphi_{1..n}\psi_{1..m} \vdash \lambda v_{1..m} (g v_{1..m} x) \land (h v_{1..m} (f_{n+2} x)) \quad (Cb) \]
Figure 3.3: GCG derivation: “Maotaijiu, which is titled as the Chinese national liquor, has a long history” (a) and its associated dependencies (b)
Rule Ca conjoins the conjunct to the left, propagates any non-local dependencies, and assigns dependency \( n + 1 \) to the left conjunct. Rule Cb conjoins the conjunct to the right, propagates any non-local dependencies, and assigns dependency \( n + 2 \) to the right conjunct. Rule Cc is proposed to handle conjunctions of three or more conjuncts, in which the initial conjunction lexical item is elided and coindexed with the last conjunction lexical item.

(28) 政府 支持 并 鼓励 农业

government support and encourage agriculture

‘The government supports and encourages agriculture’

In (28), the conjunction 支持并鼓励 ‘support and encourage’ takes 政府 ‘the government’ as the first argument, 农业 ‘agriculture’ as the second argument. 支持 ‘support’ is the third argument (first conjunct) of the conjunction word 并 ‘and’ and 鼓励 ‘encouragement’ is its fourth argument (second conjunct). The conjunction in (28) is analyzed as an right node raising construction (marked as ‘RNR’) in the Penn Chinese Treebank. In our GCG annotations, we simply analyze it as conjunction of transitive verbs as shown in (28).

Gapping constructions are annotated in the Penn Chinese Treebank. For example, the second conjunct 三家独资企业 ‘three foreign-owned enterprises’ in the sentence in Figure 3.4 is annotated
as a VP where, presumably, the omitted verb is coindexed with the verb occurring in the first conjunct. However, it is widely accepted that gappings are not allowed in Chinese (Tai, 1969, 1994). We therefore reannotate the gapping constructions in Penn Chinese Treebank into coordinations of noun phrases. The GCG derivation of the sentence in Figure 3.4 is shown in Figure 3.5.
3.2 Summary

This chapter describes a generalized categorial grammar formalism for Mandarin Chinese. Unlike previous categorial grammars where a small set of inference rules are used, the generalized categorial grammar formalism introduced here allows both inference rules and lexical categories to be language-specific. This modification yields a much smaller set of syntactic categories with a slightly larger set of inference rules. The inference rules stipulate the combination of different syntactic categories and interpretation of the combinations in terms of syntactic dependencies. This formalism offers language-specific analysis of some unique syntactic constructions in Mandarin Chinese, such as ba-, bei-, and de-constructions, which make substantial use of unbounded dependencies.
Chapter 4: Syntactic Parsing Evaluations

Parsing is to reveal latent syntactic structure of a language according to some grammar formalism. The latent syntactic structure of a language usually provides a straightforward interface between the syntax of the language and certain forms of meaning representations. Therefore, automatic parsing has been widely used in many natural language processing applications where deep understanding of language meaning and structure is needed, such as question answering, event detection and machine translation.

Chinese language parsing has seen great improvement as successful parsing algorithms which are proven to be effective in English are applied to syntactic parsing in Mandarin Chinese. However, the parsing performance of Mandarin Chinese lags behind English in various grammar formalisms. This research defines a grammar formalism tailored to the language-specific features of the Chinese language. We hope a grammar formalism like this can yield better learnability of the grammar in automatic parsing.

This chapter first describes how we obtain broad-coverage GCG annotations as described in Chapter 3 and how to train automatic syntactic parsers with the annotations. It then reports the results and evaluations of various parsing experiments conducted with the parsers.
Figure 4.1: Annotating relative clause structure in the sentence “Chongming has a position that is superb”
4.1 Reannotation

In order to make the grammar formalism described in Chapter 3 applicable to natural language processing tasks, we need broad-coverage GCG annotations to evaluate its generality and an automatic syntactic parser trained on the annotations to evaluate its learnability. However, annotating raw text data with the GCG annotations is not realistic because human annotations are expensive and time consuming. Many researchers make use of existing syntactically annotated corpora and convert them into the target representations by rewriting rules. For example, Tse and Curran (2012) converted Penn Chinese Treebank into the Chinese CCGbank and Nguyen et al. (2012) converted the Penn English Treebank into English GCG annotations. Therefore we chose to reannotate the Penn Chinese Treebank into the GCG annotations.

We use a set of reannotation rules similar to those described by Nguyen et al. (2012) to reannotate the Penn Chinese Treebank into GCG trees. These reannotation rules work within a perl script that traverses each bracketed sentence in the Penn Chinese Treebank by selecting each pair of matching brackets from the top of the tree to the bottom, then running a sed-like pattern substitution rule on each selection. The reannotation rules are defined as recursive rewrites that progress down the Treebank trees, propagating -a, -b, -c, -d and -g arguments as they go. Figure (4.1) shows a single step in the annotation of the sentence ‘Chongming has a position that is superb.’ In Figure 4.1, the reannotation rule deletes the empty complementizer node WHNP, and annotates the IP node under CP as V-gN and the relativizer’s ‘de’ as D-aV. The node ‘V-gN’ still carries the particular trace marker, which helps locate where the trace is generated in the follow-up annotation. In general, specifiers of head projections are annotated as preceding arguments, e.g., -aN for nominal subject, and complements of head projections are annotated as succeeding arguments: e.g. -bN for objects, and -bV for sentential complements, etc. Trace annotations for relative clauses, topicalizations,
and passivizations are implemented by introducing gap arguments: \(-gN\) (for noun phrase gaps) and \(-g(R-bV)\) (for adverbial phrase gaps).

In many cases the conversion to this categorial grammar replaces Treebank category labels with more general specifications. For example, we do not distinguish noun phrases (NP) from nouns (N) in general. Proper nouns (NP), common nouns (NN), temporal nouns (NT) and pronouns (PN) are not differentiated. In other cases, the conversion to categorial grammar distinguishes categories that are conflated in the Treebank. For example, ‘IP’ category can be ‘V,’ ‘V-gN’ or ‘V-g(R-bV),’ depending on whether a gap is present in the clause. The categories of verbs are decided by their valence. For example, ‘VV’ can be ‘V-aN’ (intransitive verbs), ‘V-aN-bN’ (transitive verbs), ‘V-aN-bN-bN’ (ditransitive verbs), ‘V-aN-bV’ (verbs taking sentential complement), ‘V-aN-b(V-aN)’ (subject-control verbs) and ‘V-aN-bN-b(V-aN)’ (object-control verbs), etc.

With around 200 annotation rules, we fully annotate 73% of sentences (19,113 sentences out of 26,062) from the Penn Chinese Treebank 5 and 6.

### 4.2 Parsing Experiments

#### 4.2.1 Parsing performance with different training sets

For our parsing experiments, we choose to use the Berkeley latent-variable parser to train a probabilistic syntactic parser on the GCG annotations. We divided the fully annotated sentences into training, development and test sections according to the section divisions suggested by Tse and Curran (2012). In order to have a reference when we evaluate the parsing performance of the GCG parser, we also trained a parser on the Chinese CCGbank. The Chinese CCGbank is obtained by converting the Penn Chinese Treebank into CCG annotations according to Tse and Curran (2012).\(^2\)

We divided the fully annotated sentences in both grammars into training, development and test

\(^2\)https://github.com/jogloran/cnccgbank

46
sections according to the section divisions suggested by Tse and Curran (2012). The Chinese CCG parser is trained both on the full training set (ccg.full) and on the same training set used for training the Chinese GCG parser (ccg.same). The detailed section divisions are shown in Table 4.1.

The Berkeley PCFG grammar trainer was used in Tse and Curran (2012) to train an automatic syntactic parser for Chinese CCG annotations. Therefore, for the two CCG parsers, ccg.full and ccg.same, we use the Berkeley latent-variable PCFG trainer, with 5 split-merge cycles, which is the best setting indicated by (Tse and Curran, 2012). We use the same grammar trainer to train a parser with the Chinese GCG annotations that are reannotated from the Penn Chinese Treebank. The PCFG trainer was used ‘off the shelf’ and run with its default parameters, only varying the number of split-merge iterations on the development section. We found 5 split-merge iterations yielded the best parsing performance in the development section.

### Table 4.1: Train/Dev/Test Split

<table>
<thead>
<tr>
<th>Model</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccg.full</td>
<td>22680</td>
<td>926</td>
<td>2230</td>
</tr>
<tr>
<td>ccg.same</td>
<td>15957</td>
<td>926</td>
<td>2230</td>
</tr>
<tr>
<td>gcg</td>
<td>15957</td>
<td>926</td>
<td>2230</td>
</tr>
</tbody>
</table>

### Table 4.2: Parsing results on the development set

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>P</th>
<th>F</th>
<th>tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccg.same</td>
<td>78.64</td>
<td>78.96</td>
<td>78.80</td>
<td>85.62</td>
</tr>
<tr>
<td>ccg.full</td>
<td>80.69</td>
<td>81.13</td>
<td>80.91</td>
<td>87.24</td>
</tr>
<tr>
<td>gcg</td>
<td>82.70</td>
<td>83.86</td>
<td>83.28</td>
<td>93.65</td>
</tr>
</tbody>
</table>
Tables 4.2 and 4.3 show the parsing performance of the parsers on the development and test sets. We use standard evalb, a bracket scoring algorithm, to measure the parsing accuracy of the trained syntactic parsers. Evalb takes the parse outputs from an automatic parser and compares them with the gold annotations to report the precision (P), recall (R), F1-score (F) and tagging accuracy (Tag), as reported in 4.2 and 4.3.

The parsing results show that a larger training set is beneficial to the parsing performance of the Chinese CCG parser; the parsing performance of the CCG parser trained on the full training set performs substantially better than the parser trained on 73% of the training set. The GCG parser, trained on 73% of the training set, seems to parse reasonably well. However, direct comparison of the parsing performance of these two parsers is not legitimate here because these two grammars define different categories and different tree structures. We do not conduct a hypothesis test here for the difference in parsing performance because the two formalisms are different, and accuracy results can be biased against the representation with the more fine-grained set of categories.

### 4.2.2 Parsing performance comparison with CCG

In order to evaluate the Chinese GCG annotations in terms of parsing accuracy, we compare the parsing performance of a latent-variable parser trained on Chinese GCG annotations with that of the same parser trained on Chinese CCG annotations. To ensure a fair comparison between these

---

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>P</th>
<th>F</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccg.same</td>
<td>78.39</td>
<td>78.55</td>
<td>78.47</td>
<td>85.02</td>
</tr>
<tr>
<td>ccg.full</td>
<td>79.77</td>
<td>79.93</td>
<td>79.85</td>
<td>86.33</td>
</tr>
<tr>
<td>gcg</td>
<td>82.19</td>
<td>83.07</td>
<td>82.63</td>
<td>93.66</td>
</tr>
</tbody>
</table>

Table 4.3: Parsing results on the test set
Figure 4.2: Constructing common test set
Table 4.4: Parsing results, error reduction ratios and significance testing results on the common test set of NoUnary+NoLab trees.

<table>
<thead>
<tr>
<th></th>
<th>% Err. Reduct. vs.</th>
<th>p-value vs.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1</td>
<td>ccg.same ccg.full</td>
</tr>
<tr>
<td>ccg.same</td>
<td>88.76</td>
<td>–</td>
</tr>
<tr>
<td>ccg.full</td>
<td>89.39</td>
<td>–</td>
</tr>
<tr>
<td>gcg</td>
<td><strong>90.07</strong></td>
<td>11.65</td>
</tr>
</tbody>
</table>

two grammars, we test the parsing performance of these two grammars on a common test set of sentences to which the two grammars assign the same tree structure when syntactic labels and unary branches are removed, see Figure 4.2.

To construct the test set, we first remove all unary branches from the gold and predicted annotations in the test set for each of the two grammars, using the upper category as the category of the final merged node to get a completely binary tree (NoUnary), as shown in Figure 4.2b. Then we additionally relabel all categories with ‘X’ (NoLabel), as shown in Figure 4.2c, to remove any bias in evaluation due to one grammar having a larger number of categories than the other.

Then we use as our final test set those sentences for which both grammars define the same unlabeled binary branching tree structure. We found 984 sentences which have exactly the same unlabeled binary structures in both grammars in the test section proposed in Table 4.1.

Table 4.4 shows the parsing results (F1) on parses with both syntactic category labels and unary branches removed (NoUnary+NoLab). After removing unary branches, the parses have exclusively binary tree structures and have identical results for precision, recall and F1 in parsing evaluations. Since both grammars predict exactly the same binary tree structures with exactly the same (‘X’) categories, it is legitimate to perform a significance test to compare the parsing performances. Results in Table 4.4 show that the parser trained on the GCG-annotated corpus is more accurate.
with strong significance \( p < 0.001 \) than the same parser trained on the CCG-reennotated corpus of the same size. We also observe a significant improvement \( p < 0.05 \) of the GCG parser over the CCG parser which is trained on a larger training set.

4.3 Construction Specific Parsing Performance Analysis

As shown in Chapter 3, several syntactic constructions, such as the \( ba \) construction, passive voice, and topicalization, are analyzed differently in these two representations. In order to better understand the parsing performance difference shown in Table 4.4 between the Berkeley-trained Chinese GCG grammar and CCG grammar, we want to know whether different analyses of these constructions are responsible for the difference in parsing performance we have observed.

In the following experiment, we choose the \( ba \) construction to examine the effect of different analyses on parsing performance, since the \( ba \) construction is commonly used in Mandarin Chinese and more likely to show a significant effect on the overall parsing performance.

Tse (2013) proposes two syntactic categories for the particle \( ba \) and argues that the \( ba \) construction has gapped and non-gapped configurations.

\[
gapped \text{ba} \mapsto ((S[dcl]\NP_a)/TV_{a,p})/NP_p
\]

\[
\text{non-gapped \text{ba}} \mapsto ((S[dcl]\NP_a)/(S[dcl]\NP_a))/NP
\]

In the gapped \( ba \) construction, the nominal complement of the particle \( ba \) is co-referent with the object of the verb phrase. In the non-gapped \( ba \) construction there is no gap in the verb phrase, which means that there is no dependency relation between the noun phrase after \( ba \) and the verb. These two cases can be illustrated by (32a) and Figure (4.3b) in CCG annotations.

It is true that some \( ba \) constructions have their equivalent sentences in canonical word order, but some do not. For example, (20) can be expressed in (29), while we cannot rephrase (21) as (30).
(29) 猫 吃了 鱼
cat ate fish
‘The cat ate the fish.’

(30) *我们 建成 崇明 港口
we build-into Chongming port
Intended: ‘We built Chongming into a port.’

However, we do not think this is because there are two kinds of ba constructions and we need two different syntactic categories for the particle ba. We think the difference we observe here is attributable to the different verbs used here. More specifically, we think the verb 吃 ‘eat’ and the verb 建成 ‘build-into’ are two different types of verbs in Chinese. As shown in (20), 吃 ‘eat’ has category V-aN-bN, which is a canonical category for a transitive verb. The syntactic category for 建成 ‘build-into’ is B-aN-aN-bN, which means this verb needs to take three noun phrases as arguments and it has to be a verbal complement of the ba particle. Note that in our GCG analysis, Chongming is still an argument of the verb 建成 ‘build-into’, even though it occurs on the left of the verb.

In our analysis, the ba particle does not take the fronted noun phrase as a complement as in the Chinese CCGbank. On the contrary the ba particle takes the noun phrase and the verb together as complement. This analysis is motivated by sentences like (31).

(31) 他 把 饭 吃了, 碗 洗了, 就 睡觉了
he BA meal ate, dish washed, then slept
‘He ate his meal, washed the dishes and went to sleep.’

In (31), the coordination of 饭吃了 ‘meal ate’ and 碗洗了 ‘dish washed’ works together as a complement of the particle ba, which suggests 饭吃了 ‘meal ate’ and 碗洗了 ‘dish washed’ should be analyzed as a constituent before they conjoin and then combine with the particle ba.

Even though we have a very different analysis of the ba construction from the Chinese CCGbank, we do not try to argue that our analysis generates more reasonable dependency interpretation for
ba constructions. We believe the expected dependencies can be extracted from derivations in both analyses as long as the annotations are consistent. The purpose of the following experiment is to show which analysis can generalize better when learned in an automatic parser, for which we still choose to use the Berkeley PCFG grammar trainer (Petrov and Klein, 2007).

We map the analysis of the ba construction in the Chinese CCG Treebank into GCG annotations. We make sure that we only change the categories and branches that are affected by the ba construction and keep the rest of the tree the same as the original GCG tree. We use a sentence from the development set of the Chinese Penn Treebank to illustrate how to map the CCG analysis of the ba construction into GCG annotations in Figure 4.3. We want to be clear here that we do not agree with CCGbank’s analysis of the sentence in Figure 4.3.

In Figure 4.3a, we provide the GCG derivation of the sentence. The corresponding derivation in CCG is shown in Figure 4.3b. The we map the affected CCG categories into GCG categories in a heuristic way: the backward application operator \ is mapped to -b and the forward application operator / to -a; NP in CCG is mapped to N in GCG and S or Sdc1 is mapped to V. Figure 4.3c shows the derivation resulting from this mapping, in which the CCG category for ba, ((Sdc1\NP)/(Sdc1\NP))/NP is mapped into the GCG category V-aN-b(V-aN)-bN, and the CCG category for VP Sdc1\NP is mapped to V-aN in GCG.

There are 434 ba sentences in the GCG training set, which is about 3% of the whole training set. We changed the analysis of the ba construction in these 434 sentences into CCG analyses of ba and trained a Berkeley Parser with the converted training set. We then tested the parser on the development set and compared the parsing results with those obtained by the parser trained on the original GCG training set, where the ba construction is annotated with the GCG analysis.

Table (4.5) shows the parsing results with two different analyses of the Ba construction. The parsing results are worse when we parse the development set with the grammar trained on the
Figure 4.3: Derivations of the sentence *We will push the reform as a whole to depth.*
Table 4.5: Parsing results on the development set where *ba* sentences are annotated with either CCG analysis (*ccg.ba*) or GCG analysis (*gcg.ba*)

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>P</th>
<th>F</th>
<th>tag</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>ccg.ba</em></td>
<td>82.21</td>
<td>83.52</td>
<td>82.86</td>
<td>93.10</td>
</tr>
<tr>
<td><em>gcg.ba</em></td>
<td>82.70</td>
<td>83.86</td>
<td>83.28</td>
<td>93.65</td>
</tr>
<tr>
<td>p-value</td>
<td>0.04</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

sentences with CCG analysis of the *ba* construction. We can see that the reduction in recall is statistically significant at the level of 0.05. Considering there are only 17 *ba* sentences in the development set (around 2% of the development set), this change of parsing performance is rather remarkable.

We manually inspected the 17 *ba* sentences in the development set and found that the *ba* particle had three different CCG categories. One of the category is \((Sdcl\ NP)/(Sdcl\ NP)/NP\), as shown in Figure 4.3b, the other two are shown in (32)\textsuperscript{4}.

(32) Other Categories for *ba* particle in CCG grammar

a. *ba* as \((Sdcl\ NP)/(Sdcl\ NP)/NP\)\/NP
   ‘move the statues away’

\[
\frac{\text{把}}{\text{(Sdcl\ NP)/(Sdcl\ NP)/NP}} \quad \frac{\text{塑像}}{\text{NP}} \quad \frac{\text{搬走}}{(Sdcl\ NP)\/NP} \quad \frac{\text{move-away}}{\text{Fa}}
\]

\[
\frac{\text{NP}}{(Sdcl\ NP)/(Sdcl\ NP)/NP)}/\text{Fa}
\]

b. *ba* as \((Sdcl\ NP)/Sdcl\)
   ‘put learning Securities Law on agenda’

\textsuperscript{4}For simplicity, only the part of sentence where *ba* construction occurs is shown in the following examples.
For comparison, in (33) we show the GCG derivations of the phrases in (32).

(33) GCG derivations for the corresponding phrases

a. ‘move the statues away’

b. ‘put learning Securities Law on agenda’

As we see in Figure 4.3 and (32), the *ba* particles are receiving different categories. In particular, the *ba* constructions in Figure 4.3b and (32a) are not syntactically distinguishable, which will make it hard for the parser to learn. Out of the 17 *ba* sentences in the development set, 4 sentences had wrong predictions of the *ba* particle by the parser trained with the CCG *ba* analysis. The parser did not correctly predict the category of the *ba* particle in (32b). The remaining three cases are all caused by the confusion between *ba* as ((Sdcl\NP)/((Sdcl\NP)/NP))/NP or ((Sdcl\NP)/(Sdcl\NP))/NP, which as we commented above, is not distinguishable in many cases. In contrast, since the *ba* particle is consistently annotated as V-aN-b(B-aN) in the GCG analysis, there is no mistake in predicting the category of the *ba* particle for GCG grammar.
4.4 Summary

This chapter introduced the reannotation process from which a broad-coverage corpus of GCG annotations were obtained. Then we conducted a series of parsing experiments to evaluate the learnability of this grammar formalism and compared the results with those from Chinese CCGbank.

Our experimental results show that the Berkeley latent-variable parser trained on the Chinese GCG parser annotations yields significantly better parsing results than that trained on the Chinese CCGBank when predicting a common test set which has the same unlabeled binary tree structures in the gold annotations of both grammars. We believe that the Chinese CCG parser suffers more from data sparsity effects than the Chinese GCG one. Excluding those words which are only associated with one preterminal category, the lexical-categorial confusion rate is 3.45 for the Chinese CCG annotations and 2.59 for the Chinese GCG annotations, which can also be inferred from the large gap (more than 5 points) between their tagging accuracies. Also different analysis of some syntactic constructions might also contribute to the difference in parsing performance we observed, as shown in Section 4.3.

Finally, all the parsing experiments reported here are conducted by using the Berkeley parser. We do not exclude the possibility that other grammar training algorithms can handle sparse data problems better.
Chapter 5: Semantic Dependency Parsing Evaluations

Syntactic parsing evaluation can only be conducted among parsers which compete to predict the same target syntactic representations. In Chapter 4, we selected from the original test set a subset of trees where the unlabeled binary-branched trees are shared by the two grammar formalisms to test the parsers’ performance. This subset of the test data can be biased because they are not randomly selected from the corpus. Evaluating grammar formalisms by examining how well they can predict the gold-annotated semantic dependencies can be a way to avoid this difficulty.

Although syntactic representations can be used to infer semantic dependencies, semantic dependencies are more abstract than grammatical relations in the way that the arcs in semantic dependency represent some type of semantic relation, rather than grammatical relations. Semantic dependencies can come with a graph structure, rather than a tree structure which is assumed by most syntactic representations. Therefore, semantic dependency representations can sometimes provide more direct applications to natural language processing tasks such as question answering, event detection and information extraction.

In this chapter, we test how well the Chinese GCG grammar can help to predict Chinese semantic dependencies manually annotated in a shared task.
5.1 Chinese Semantic Dependency Annotations

We chose to participate in SemEval 2016, Task 9: Chinese Semantic Dependency Parsing (Che et al., 2016), to test how well the Chinese GCG parser helps to recover these semantic dependencies. The Chinese semantic dependencies used in this task are extended from the tree-structured dependencies in the previous shared task of Chinese semantic dependency parsing (Che et al., 2012) to the form of directed acyclic graphs, as shown in Figure 5.1. The annotators of the corpus found that the tree-structured dependencies are not sufficient to account for the latent semantic relations in some Chinese constructions (Che et al., 2016).

As shown in Figure 5.1, the semantic dependency annotations have directed arcs connecting pairs of words in the sentence and each arc is labeled with a semantic dependency type. There are totally 156 semantic labels annotated in the corpus. The task provides two corpora from two different domains, NEWS and TEXTBOOK (from textbooks used in elementary schools). The detailed statistics of the corpora could be found in Table 5.1
Table 5.1: Distribution of the corpora

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEWS</td>
<td>#sent</td>
<td>8,301</td>
<td>534</td>
</tr>
<tr>
<td></td>
<td>#word</td>
<td>250,249</td>
<td>15,325</td>
</tr>
<tr>
<td>TEXT</td>
<td>#sent</td>
<td>10,817</td>
<td>1,546</td>
</tr>
<tr>
<td></td>
<td>#word</td>
<td>128,095</td>
<td>18,257</td>
</tr>
</tbody>
</table>

Table 5.2: Data format of SemEval 16, Task 9

<table>
<thead>
<tr>
<th>ID</th>
<th>FORM</th>
<th>POS</th>
<th>FEAT</th>
<th>HEAD</th>
<th>REL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>现在</td>
<td>‘now’</td>
<td>NT</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>她</td>
<td>‘she’</td>
<td>PN</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>她</td>
<td>‘she’</td>
<td>PN</td>
<td>-</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>脸色</td>
<td>‘face’</td>
<td>NN</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>苍白</td>
<td>‘pale’</td>
<td>VA</td>
<td>-</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>,</td>
<td></td>
<td>PU</td>
<td>-</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>好像</td>
<td>‘seems’</td>
<td>AD</td>
<td>-</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>生病</td>
<td>‘sick’</td>
<td>VV</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>了</td>
<td>‘already’</td>
<td>SP</td>
<td>-</td>
<td>7</td>
</tr>
</tbody>
</table>

We can see although the TEXTBOOK corpus contains more sentences, it has many fewer words, which means sentences in the TEXTBOOK corpus are on average much shorter than those in the NEWS corpus.

The data comes in the CoNLL format: each word takes one line and sentences are separated by an empty line. For each word, five fields of total ten fields are filled with token ID, token form, its part-of-speech tag, its head ID and the semantic relation label. An example is shown in Table 5.2. Note that when a word has multiple heads, it appears in multiple consecutive lines. For example, 她
'she' in Table 5.2 has two lines, corresponding to the two heads, 脸色 'face' and 生病 'sick', with which 她 'she' is associated in the sentence.

### 5.2 GCG-based semantic dependency parsing system

Although semantic dependencies have the advantage of abstracting away a specific syntactic formalism and being able to serve as the evaluation target for various parsing systems, different semantic dependency representations have their own definitions about what grammatical relations are counted as semantic dependencies. The semantic dependencies annotated in the Chinese semantic dependency parsing task are different from the syntactic dependencies read off our GCG derivations in several ways. Adhering to the principle that the semantic dependencies should be established between content words, the semantic dependency schema used in the task handles function words differently from the GCG syntactic dependencies. For example, prepositional phrases are annotated in the way that the preposition word is treated as a function word or a non-content word in the Chinese semantic dependencies. The GCG syntactic dependencies, however, do not distinguish function words and content words. Dependencies are constructed by the syntactic combinations stipulated by the inference rules without considering whether some word is a function word or a content word.

(34) 在 教室 看 书
at classroom read books

'Read books at the classroom'

a. Syntactic dependencies in GCG

```
在 at 教室 classroom 看 read 书 book
R-bV-bN N Ab V-aN-bN V-aN-bN Ab
```
b. Semantic dependencies in SemEval 2016

For example, 看 ‘read’ and 教室 ‘classroom’ are connected in the semantic dependencies of the shared task as shown in (34b). These two words, however, are not directly connected syntactically according to the GCG derivation in (34a).

Another difference is that the semantic dependencies have many more dependency or relation labels than the syntactic dependencies defined in the GCG derivations. The GCG syntactic dependencies only have a few labels to label the positions of the argument, e.g.: ‘1’ as the first argument, ‘2’ as the second argument, and so on, as shown in (34a). The dependency labels in the semantic dependencies contain more specific content. For example, a subject could be an experiencer (Exp), an agent (Agt), an affecter (Aft) or a possessor (Poss); an object could be a patient (Pat), content (Cont), a product (Prod), or an origin (Orig), etc. Therefore the semantic dependencies are different from the GCG syntactic dependencies both in arcs and in labels and there is no direct match between the semantic dependencies and syntactic dependencies.

In order to predict the semantic dependencies in the task based on the GCG syntactic dependencies, we proposed a Chinese semantic dependency parsing system which consists of two components: a parser trained using the Berkeley Grammar Trainer on the Penn Chinese Treebank reannotated in a Generalized Categorial Grammar, and a multinomial logistic regression classifier which makes use of syntactic features obtained from the parser and lexical features to predict the dependency labels.
5.2.1 Parsing training data

We first trained a GCG syntactic parser using a latent-variable PCFG trainer (Petrov and Klein, 2007) on GCG annotations of 19,113 sentences from the Penn Chinese Treebank 5 and 6. We then use the parser to parse sentences from the training sets in both the NEWS and TEXTBOOK domain to obtain GCG derivations of the these sentences. Then we extracted syntactic dependencies and other syntactic features from the derivations to use them as input for the multinomial dependency label classifier.

5.2.2 Multinomial Dependency Label Classifier

We chose to train a multinomial logistic regression classifier to predict the dependencies and their labels based on the syntactic features provided by the parser and other lexicon features. There are more sophisticated machine learning algorithms for a classification task such as this one. We chose to use a more simpler algorithm in order to observe the effect of the syntactic features better.

The task of this multinomial classifier is to first detect whether there is a dependency between pairs of words, and if there is a dependency, determine the dependency label. Ideally we should examine all pairs of words in a sentence for possible directed dependency relations. However, the search space for a sentence of $n$ words is $O(n^2)$, while usually there are only around $2n$ dependencies in a sentence of length $n$ words, which means pairwise searching will lead to very unbalanced classification. In the current case, pairwise searching leads to more than 95% of pairs of words holding no dependency relations and less than 5% having dependency relations of more than 150 categories.

In order to get around this difficulty, we only examine a subset of pairs of words which are more likely to hold dependency relations. We first examine all pairs of words which are identified by the GCG parser to have syntactic dependencies between them. Then we noticed that
the dependencies identified by the parser sometimes have different directions than the semantic
dependencies annotated in the task. Therefore, for each dependency identified by the parser, we
also add a dependency which has the reverse direction. For example, if the parser predicts that a
dependency such as \( (eat, cat) \), in which the head is \( eat \), the dependent is \( cat \), the dependency label
is ‘1’, we would add another dependency with inversed direction: \( 1\text{-inv}(cat, eat) \). By doing so, we
can increase the coverage of the annotated semantic dependencies to around 83% in the training set.

There are totally 156 semantic dependency labels used in the task. Since the classifier also needs
to decide whether a dependency relation exists between each pair of words, we add a “NoRel" label for those pairs of words which, according to the gold annotation, do not hold any dependency
relation between them. Therefore, there are totally 157 dependency labels to predict.

We train a one-vs-all multiclass classifier from the Vowpal Wabbit machine learning package
due to its fast learning speed and friendly interface.\(^5\) We used the following features to predict the
dependency labels.

- **Lexical features**: the 300-dimensional word embeddings of the head and dependent words
  trained with \textit{word2vec} (Mikolov et al., 2013) on the full Chinese Wikipedia,\(^6\) the Chinese
  Gigaword and the training and development datasets in this task;

- **POS features**: the 50-dimensional vector representations of the POS tags of the head and
  dependent trained with the POS tag sequences from training and test datasets in this task;

- **Linear distance**: the linear distance of the head and the dependent in the sentence;

- **Path distance**: the distance of the nodes of the head and the dependent in the output syntactic
tree of the sentence from the syntactic parser;

\(^5\)https://github.com/JohnLangford/vowpal_wabbit/wiki
\(^6\)https://zh.wikipedia.org
- **Syntactic categories**: the GCG syntactic categories of the head and the dependent, such as ‘V-aN-bN’ for transitive verbs or ‘N’ for noun phrases;

- **Pred-arg dependency labels**: the dependency labels predicted by the parser, such as ‘1’ for the first argument or ‘2’ for the second argument;

- **Repetition penalty**: the reciprocal of the number of heads that the dependent word has, to penalize proposing too many heads for one word;

- **Joint features**: two-way combinational features between GCG syntactic categories of the two words and the dependency label, such as “V-aN_1” or “N_1”;

### 5.3 Results

This Chinese semantic dependency parsing task comes in two domains, the newspaper articles (News) and texts selected from Chinese textbooks (Text). In our experiment, we found the combining the two training sets yields better accuracy for the textbook corpus and a slightly worse performance for the newspaper corpus. Therefore the News results reported in Table 5.3 and 5.4 are obtained by a classifier only trained on the newspaper corpus, and the Text results are obtained by a classifier trained on the combined training set of the newspaper corpus and the textbook corpus.

<table>
<thead>
<tr>
<th></th>
<th>LF</th>
<th>NLF</th>
<th>UF</th>
<th>NUF</th>
</tr>
</thead>
<tbody>
<tr>
<td>News</td>
<td>58.99</td>
<td>54.99</td>
<td>76.64</td>
<td>63.51</td>
</tr>
<tr>
<td>Text</td>
<td>65.31</td>
<td>56.74</td>
<td>78.19</td>
<td>66.61</td>
</tr>
</tbody>
</table>

Table 5.3: Results of development set where LF is the F1 score of the labeled dependency, NLF is the F1 score of the non-local dependency, UF is the F1 score of the unlabeled dependency and NUF is the F1 score the non-local unlabeled dependency.
The results in Table 5.3 show that newspaper text is more difficult to parse, even though the GCG parser is trained on a newspaper corpus. However, it also shows that the parser trained on the newspaper corpus can generalize nicely to another domain such as the textbook corpus, where more diverse syntactic constructions are found. Non-local dependencies are fuzzily defined as those dependencies which connect multiple heads to one word in the evaluation of this shared task (Che et al., 2016). The underlying assumption is that when there are multiple heads associated with one word, some of them are likely to be syntactically nonlocal. Traditional semantic parsers which assume tree structured dependencies might find it extremely challenging to recover these types of dependencies. That was probably another reason why the organizers of the task list F1 score of nonlocal labeled dependency (NLF) and F1 score of nonlocal unlabeled dependency (NUF) as evaluating metrics in Table (5.3), although those are nonlocal dependencies are very vaguely defined.

Table 5.4 shows the results on the test set compared with the system yielding the highest F1 score for the labeled dependencies. We can see that the current system is around 3 percentage lower than the top system in terms of the labeled F1 score. Considering the fact that the parser is not directly trained on the task-specific dependency annotations, these results look reasonably good with the rather simplistic machine learning architecture. We can see from Table 5.4 that the

<table>
<thead>
<tr>
<th></th>
<th>LF</th>
<th>NLF</th>
<th>UF</th>
<th>NUF</th>
</tr>
</thead>
<tbody>
<tr>
<td>News</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCG</td>
<td>55.69</td>
<td>49.23</td>
<td>73.72</td>
<td>60.71</td>
</tr>
<tr>
<td>TOP</td>
<td>58.78</td>
<td>40.84</td>
<td>77.64</td>
<td>60.20</td>
</tr>
<tr>
<td>Text</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCG</td>
<td>65.17</td>
<td>54.70</td>
<td>78.83</td>
<td>65.71</td>
</tr>
<tr>
<td>TOP</td>
<td>68.59</td>
<td>50.57</td>
<td>82.41</td>
<td>64.58</td>
</tr>
</tbody>
</table>

Table 5.4: Results of the test set compared with the best system where GCG is the current system and TOP is the system with the best labeled F1 score.
dependencies of words having multiple heads are indeed more challenging. In both our system and the top system, the F1 scores are lower for these types of dependencies. However, Table 5.4 shows that our system achieves the best performance on multi-headed dependencies according to the official evaluation.

5.4 Error Analysis

In order to better understand the errors our system made in predicting the semantic dependencies, we randomly inspected around 20 sentences from each domain where the predictions of the current system are different from the gold annotations to examine the reasons, and we identified the following sources of errors.

• **Parsing errors**

  Parsing errors caused around half of the wrong predictions we inspected. One type of mistake that we noticed is that the parser often makes wrong predictions about the internal structure of complex noun phrases. For example, for the noun phrase 国际货币基金组织 ‘International Monetary Fund Organization’, the parser proposes all first three nouns to be modifiers of the head noun 组织 ‘organization’ as shown in Figure 5.2a, while 货币 ‘monetary’ actually modifies 基金 ‘fund’ in gold annotations, as shown in Figure 5.2b. Correct parsing of compound noun phrases usually requires lexical and semantic information, which could be a challenge for syntactic parsers.

• **Uncovered linguistic phenomena**

  There are some linguistic phenomena that are annotated in the semantic dependencies but not represented by the syntactic dependencies. For example, the gold dependency annotations issued by the task contain dependencies involving co-reference, as shown in (35a) and (35b).
Figure 5.2: Tree structures for 'International Monetary Fund Organization'
(35) a. 我觉得 自己是 世界上最幸福的人了。
   I think myself is in-the-world happiest DE person ASP.
   ‘I think I am the happiest person in the world.’

   b. 鲁肃问：你叫 我 来 做什么？
   Lusu asked him: “you asked me come do what?”
   ‘Lusu asked him: “Why did you ask me to come?”’

In (35a), 我 ‘I’ is annotated to have an eEqu relation with 自己 ‘myself’. In (35b), 鲁肃 ‘Lusu’ has an eEqu relation with 我 ‘me’. Dependencies like these, especially the one in (35b), cannot be resolved easily by a syntactic parser. In order to recover this type of dependencies, we might need an extra layer of post-processing to do co-reference inference based on discourse information.

• Ambiguous constructions

In some cases, a sentence can be analyzed in more than one way. All of them are reasonable analyses but each gives different dependencies.

(36) 我爱他有志气
   I love he have aspiration
   ‘I love that he is ambitious’ or ‘I love him for being ambitious.’

(37) 这里一定有 人 来 过
   here must have people come TENSE
   ‘Someone must have come here before’

The current parser parses both (36) and (37) as object control sentences. In (36) 他 ‘he’ is the object of 爱 ‘love’ and the subject of 有志气 ‘have aspiration’. In the gold annotation of (36), 爱 ‘love’ takes a sentential complement 他 有志气 ‘he has aspiration’. Therefore, for (36), our system proposes a dependency between 爱 ‘love’ and 他 ‘he’, while in the gold annotations, the dependency is between 爱 ‘love’ and 有 ‘have’. In (37), our parser parses 人
‘person’ as the object of 有 ‘have’ and the subject of 来过 ‘came’. 有 ‘have’, therefore, is the root of the sentence. In gold annotations, 来 ‘come’ is annotated as the root of the sentence.

- **Inconsistent annotations**

There are some cases where our predictions are systematically different from the gold annotations. For example, the current system is consistently different from the gold annotations on the identification of root where some adverbial clauses are involved.

(38) 万一 明天 下雨， 则 推迟 野游 日期
If tomorrow rain, then postpone trip date
‘If it rains tomorrow, postpone the date of trip’

(39) 既然 她 讨厌 伦敦, 为什么 他 还 在 那里 买 了 房子 ？
Given she dislike London, why he still at there buy ASP house ？
‘Given that she does not like London, why did he buy a house there?’

We think both (38) and (39) contain a conditional sub-ordinate clause, and the root of the sentence should be 推迟 ‘postpone’ in (38) and 买 ‘buy’ in (39). In gold annotations, 下雨 ‘rain’ is annotated to be the root of (38) and 讨厌 ‘dislike’ the root of (39). Our assumption is that those clauses are handled as conjunctions in gold annotations, and the first conjunct is treated as the head in the conjunction dependency.

Our predictions also do not agree with the gold annotations in some relative clauses.

(40) 这 正是 杰里米 喜欢 做 的 事情
This is Jimmy like do DE thing
‘This is exactly what Jimmy likes to do’

For (40), our system predicts that 事情 ‘things’ holds a dCont relation, i.e., the reverse of the content relation’ with the verb 做 ‘do’ as shown in Figure 5.3a. However, in the gold semantic dependency annotations, 事情 ‘things’ holds a dDesc relation with 喜欢 ‘like’. We
think it is more reasonble to have a dependency between ‘做 do’ and ‘事情 thing’, rather than ‘喜欢 like’ and ‘事情 thing’ in here.

5.5 Summary

This section introduces the Chinese semantic dependency parsing system based on the predicate-argument dependencies predicted by a Berkeley parser trained on Chinese GCG trees reannotated from the Penn Chinese Treebank. This system is around 3% lower in accuracy for the overall labeled dependency prediction compared with the top system and has achieved best performance for the multi-headed dependency recovery.

With the relatively simple classification algorithm, our system achieves the best performance in resolving multi-headed dependency in the shared task, which shows that the syntactic features provided by a GCG parser are useful in resolving nonlocal dependencies. However, the shared task does not offer a direct and construction-specific examination of syntactic parsing difficulties in a
language. For example, the semantic gold dependencies include co-reference relations as a type of semantic dependency, which goes beyond what a syntactic parser can do. The task vaguely refers to the dependencies connecting one word to its multiple heads as ‘nonlocal’ dependencies and shows this type of dependency poses a challenge for current semantic dependent parsers. However, it fails to point out more specifically what kinds of syntactic constructions or linguistic phenomena are involved in these multi-headed dependencies, so as to pin down the origin of the difficulties.
Chapter 6: Nonlocal Dependency Parsing Evaluations

In this chapter we will evaluate the dependency recovery performance of the syntactic parser trained on the Chinese GCG annotations on specific nonlocal constructions. Recovering unbounded dependencies is challenging, as reported by Rimell et al. (2009) for the English language. However, it serves as an important test of an automatic parser which cannot be easily accomplished by shallow language techniques. In spite of the low frequency of some nonlocal dependency constructions, correctly resolving these dependencies is crucial in understanding the underlying predicate-argument structure of a sentence.

Although trace categories are annotated in Penn Treebank, few state-of-the-art constituent parsers make use of these annotations to make predictions of the nonlocal dependencies. Categorial grammar frameworks are advocated partly for their well-defined representations of filler-gap phenomena in natural languages. Parsers trained on categorial grammar annotations are found to yield superior performance in recovering nonlocal dependencies in English (Nguyen et al., 2012; Rimell et al., 2009).

In this research we focus on various nonlocal dependency types in Mandarin Chinese. We make full use of the trace categories annotated in the Penn Chinese Treebank (Xue et al., 2005) to generate test sets for eight nonlocal dependency constructions. We evaluate the nonlocal dependency recovery performance of parsers trained on generalized categorial grammar annotations with the test sets of the eight nonlocal constructions to understand the relative difficulties of parsing them. We
compare a few widely used dependency formalisms for their coverage of the nonlocal dependencies annotated in our test sets, and evaluate several widely used parsers on their ability to recover nonlocal dependencies also.

6.1 Empty categories in Penn Chinese Treebank

In order to reliably annotate nonlocal dependencies, we need to know the location where the gap is introduced, i.e., the extraction site, and where the filler is. Therefore, in the current study, nonlocal dependency constructions are those constructions which contain trace categories and their fillers are either clearly marked in the Treebank annotations or can be reliably inferred from the syntactic construction.

Co-indexing is used in the Penn Chinese Treebank to indicate relationships between an empty category and an overt constituent within the tree structure. Only intra-sentential relations are annotated and coreference between overt pronouns and their antecedents are not annotated. Each constituent has an identity-index which can be an arbitrary number attached to the syntactic category of the constituent. An empty category might carry a referring index matching to some other constituent in the sentence. Note that not all empty categories are matched to some constituent, as we will see later. All empty categories have the same part-of-speech tag -NONE-. Figure 6.1 shows a sentence of passive voice from the Penn Chinese Treebank. The empty category carries a referring index ‘3’, matching it to the subject of sentence carrying the identity index ‘3’.

There are seven types of empty categories annotated in the Penn Chinese Treebank (Xue and Xia, 2000; Xue et al., 2005) as shown in Table 6.1. Among them, *OP* are annotated for the empty specifier position of CP in relative clauses, which in English are usually occupied by the relative pronouns like ‘that’, ‘which’, ‘who’, etc. In Chinese, this syntactic position is always empty and cannot be filled with any overt constituent. We, therefore, do not include *OP* into our
Figure 6.1: Treebank annotations for "The US trade deficits with China have been greatly exaggerated."

Consideration of empty categories. Another empty category we exclude from the current study is *?. *? is used to label those random empty categories which do not belong consistently to any syntactic construction. The occurrence of *? is rare and syntactically unpredictable. The other five types of empty categories and the syntactic constructions that they occur in are introduced below.

- *T* as the trace of Ā movement

Since the Penn Chinese Treebank annotations assume movement theories for discontinuous constituents, they distinguish different types of movement when annotating empty categories. *T* is used to annotate an empty category resulting from an Ā movement in a transformational grammar setting. The *T* empty category always carries a referring index that corresponds to the identity index of some constituent in the sentence.
| *T*   | trace of $\bar{A}$ movement | topicalization and object preposing constructions |
| *     | trace of A-movement          | raising and passive constructions               |
| *pro* | —                            | dropped subjects or objects                     |
| *PRO* | —                            | control structures                              |
| *OP*  | —                            | the empty operator in relative constructions    |
| *RNR* | —                            | right node raising                              |
| *??*  | —                            | other unknown empty categories                  |

Table 6.1: Empty categories in the Penn Chinese Treebank

The *T* empty category coindexes with the relative operators in various types of relative clauses. The *T* empty category can occur in an object position in an object relative clause as shown in Figure 6.2, or it can occur in a subject position in a subject relative clause.

The Penn Chinese Treebank also annotate the *T* empty category in a topic position in a topic relative clause, as shown in Figure 6.3 and an adjunct positive for a relative clause relativizing an adjunct, as shown in Figure 6.4.

The *T* empty category also coindexes with the operator in the long-bei construction. Figure 6.5 shows a *T* empty category coindexes with an operator in a long-bei construction, in which the *T* empty category coindexes with the operator.

Topicalized NPs and quotations that occurs before the ‘saying’ verb are regarded as fronted arguments and also leave a *T* empty category, as shown in Figure 6.6.

- * as the trace of A movement

The trace of A-movement is annotated as * in raising or short-bei constructions, as shown together in Figure 6.7.
Figure 6.2: Treebank annotations for 外商投资企业获得的人民币贷款 ‘The loan in Chinese RMB that foreign invested enterprises obtained’

Figure 6.3: Treebank annotations for 投资额较大的台资企业 ‘The Taiwan enterprises whose investment amounts are comparatively large’
Figure 6.4: Treebank annotations for 中国经济最为活跃的地区 ‘The areas where the economy is most active in China’

Figure 6.5: Treebank annotations for 张三被大学录取 ‘Zhangsan was admitted by a university.’
Figure 6.6: Treebank annotations for “我们会赢的”，张三说 “We will win.” said Zhangsan’

Figure 6.7: Treebank annotations for 张三好像被打 了 ‘Zhangsan seems to have been beaten’
In Figure 6.7, the *empty category at the object position of the verb ‘beat’ is coindexed with NP-SBJ-2 in the passive voice construction. NP-SBJ-2 contains another *empty category which is coindexed with the NP-PN-SBJ1 in a raising construction.

- ***PRO* in control constructions**

The *PRO* empty category is used to annotate the empty category in control constructions. The *PRO* empty categories are in complementary distribution with lexical NPs and can only occur at a subject position. The *PRO* empty categories can have a generic referent or a definite referent.

Figure 6.8 shows an example of the *PRO* empty category of generic referent. The subject of the verb 发展 ‘develop’ is non-finite and the verb itself cannot be modified by aspect-markers such as 着 (progressive aspect marker), 了 (perfective aspect marker) or 过 (experiential aspect marker).
Figure 6.9 shows a subject control construction, in which the subject of the subordinate verb 发展‘develop’ is co-referential to the subject of the subject-control verb 设法‘try’.

Figure 6.10 shows a object control construction, in which the subject of the subordinate verb 吃‘eat’ is co-referential to the object of the object-control verb 请‘invite’.

The *PRO* empty categories are not annotated to co-index with other NPs in the tree, since the coindexation is determined by the control verbs in control constructions and the referent is non-definite in a generic case.

• *pro* in pro-drop constructions

The *pro* empty category is used to annotated the pro-dropped subjects in the Penn Chinese Treebank. Different from the *PRO* empty categories, the *pro* can be replaced by overt NP lexical items. They are used to mark the omitted subjects in discourse, imperative subjects
or subjects in existential sentences. The *pro* empty categories are not co-indexed with any NPs in the trees, as shown in Figure 6.11.

- *RNR* in transitive verb coordination

*RNR* is used to annotate the omitted object when two verbs are coordinated. As shown in Figure 6.12, the omitted object of 引进 ‘introduce’ is annotated as an *RNR* empty category coindexed with the shared object 名贵品种 ‘precious varieties’. However, note that *RNR* empty categories annotated here in Chinese are different from the right node raising construction annotated in the English Treebank.

Among the five empty categories introduced above, the *T* and * empty categories are annotated with coindexing with other constituents in the tree. Since the annotations of these two type of empty categories provide the information of both the extraction site and the filler, the constructions
Figure 6.11: Treebank annotations for 关门 ‘Close the door’

Figure 6.12: Treebank annotations for 积极引进，精心种植名贵品种‘proactively introduce and carefully plant precious species’
involving these two types of empty categories can potentially be identified as nonlocal constructions in our experiment. The *RNR* empty categories, although they are co-indexed with another constituent, are not regarded as a nonlocal construction in our experiment, because they are the omitted object in the coordination of transitive verbs, rather than the right node raising construction known as a nonlocal construction in English. The GCG framework does not propose a gap category to account for the coordination of the transitive verbs.

6.2 Nonlocal constructions

6.2.1 Possible Nonlocal Constructions

Before we introduce the nonlocal constructions annotated for the current experiments, we briefly discuss some constructions which are commonly thought to be nonlocal constructions or closely related to nonlocal constructions in Chinese.

- Ba-constructions

We do not treat the *ba*-construction as a nonlocal construction in this research because the Treebank annotations do not consistently include a trace category in the *ba* construction, which makes it hard to decide the exact extraction site if we reannotate the *ba* construction as a nonlocal construction. Furthermore, as the example in Figure 6.13 shows, ‘Chongming’ cannot occur after the verb ‘build-into’, as a direct object usually does in Chinese. Therefore, it is not reasonable to propose a gap category at the post-verbal position. We think Chongming, as a semantically indispensable argument of ‘build-into’, has to occur before the verb, and dependencies between them are local in the *ba* construction. Since the verb ‘build-into’ canonically takes its second argument ‘Chongming’ from a preverbal position, in our GCG analysis of this sentence, we think the verb ‘build-into’ has the category ‘B-aN-aN-bN’, which takes ‘Chongming’ its second argument from the left.
• **Lian-constructions**

Treebank annotations treat *lian* as an adverbial modifier, as shown in Figure 6.14. *Lian* can modify a noun phrase as in Figure 6.14, a verb phrase, or a clause, as shown in (41).

(41)  a. 连 [去 医院] 都 要 先 说明 出身

\[lian \text{ go hospital dou need first show origin}\]

‘Even for going to hospital, your origin need to be checked first.’

b. 他 连 [自己 母亲 去世] 都 没有 返乡

\[he lian self mother pass-away dou not go-home\]

‘He did not go home even when his mother passed away.’

However, when *lian* modifies a noun phrase, the modified noun phrase occurs at a preverbal yet post-subject position, rather than its canonical postverbal position, and the Penn Chinese Treebank annotates a *T* empty category at the empty object position, as shown in Figure 6.15.

The *lian* construction shown in Figure 6.15 is annotated as a focus construction, in which the object occurs in a preverbal position rather than its canonical post-verbal position. Focus
Figure 6.14: Treebank annotation for 连美国警署都从这里引进警犬 ‘Even American police stations use police dogs from here’

Figure 6.15: Treebank annotation for 他连我都不认识 ‘He could not even recognize me’
constructions do not always contain the lian phrase. Figure 6.16 shows a focus construction without the lian phrase.

A sentence annotated to have the focus construction where a noun phrase occurs at a preverbal position rather than its canonical postverbal position, no matter whether it contains the lian phrase or not, is identified as a nonlocal construction in our research.

- **shi-cleft constructions**

  The shi-cleft construction as shown in (42a) is the subject of some argument about whether it contains an extracted subject. Unfortunately Penn Chinese Treebank does not contain a sentence with the shi-cleft construction as shown in (42a) to give us a chance to see how this commonly used construction is annotated in the Treebank annotations. Figure 6.17 shows the Treebank annotation of a sentence which is similar to the typical shi-cleft construction. However, the Treebank analyzes the sentence as a pro-drop sentence with its subject dropped and the sentence-initial shi as the main verb of the sentence. We notice that there is no trace
empty category annotated for this sentence. Proposing a pro-dropped subject for *shi* is not reasonable here because no overt NP can actually occur in that position.

How to analyze *shi* in (42a) in the GCG framework depends on whether (42b) is acceptable in Chinese.⁷ If (42b) is acceptable, *shi* in (42a) should have the category V-b(V-gN)-bN, which means *shi* first takes the noun phrase in focus and then a construction with a gap category. Then a constraint for this construction is added to make the second argument of *shi*, ‘Zhangsan’ in (42a), the filler of the gap introduced in the first argument of *shi*, 喜欢李四 ‘like Lisi’ in (42a).

If (42b) is not an acceptable sentence in Chinese, which means *shi* can only be used to focus the subject, then the category of *shi* would be V-b(V-aN)-bN. The category V-b(V-gN)-bN is preferable because we would rather overgeneralize a little than fail to interpret a grammatical sentence.

(42) a. 是 张三 喜欢 李四
   *shi* Zhangsan like Lisi
   ‘It is Zhangsan who likes Lisi’

   b. 是 李四 张三 喜欢
   *shi* Lisi Zhangsan like
   ‘It is Lisi who Zhangsan likes’

- *shi-de*-pseudo cleft constructions or free relatives

(43) 喜欢 李四 的 是 张三
   like Lisi *de shi* Zhangsan
   ‘The one who likes Lisi is Zhangsan.’

(44) 爱 吃 苹果 的 应该 去 山东
   love eat apples *de* should go Shandong
   ‘Those who love apples should go to Shandong’

⁷The native speakers of Chinese I asked have divided opinions about whether (42b) is acceptable in Chinese.
Treebank annotations for the constructions shown in (43) contain a trace category. A similar sentence from the Penn Chinese Treebank is shown in Figure 6.18. Our analysis of this construction is shown in Figure 6.19. We do not annotate this construction as a nonlocal construction because the *de*-clause 喜欢李四的 ‘the one who likes Lisi’ is treated as a nominal phrase in our analysis. We think the *de*-clause in (43) and (44) are syntactically the same. Both of them are a relative clause modifying an omitted noun and serving as a nominal phrase. Therefore, although there is a gap category in the *de*-clause, the filler is not specified linguistically. This analysis offers a unified treatment of *de*-clauses in general and also allows us to treat 是 shì simply as a transitive verb which takes two nominal phrases as its arguments.
Figure 6.18: Treebank annotation for 大量出现的是以前不曾出现的新情况 ‘what has emerged a lot was new situations that have never emerged before’

Figure 6.19: GCG analysis for 喜欢李四的是张三 ‘the one who likes Lisi is Zhangsan’
6.2.2 Annotated Nonlocal Construction

For all the empty categories annotated in the Penn Chinese Treebank, the *T* and * empty categories are annotated to be coindexed with other constituents in the sentence. Therefore, we selected nonlocal constructions that contain these two types of empty categories to annotate. More specifically, we choose topicalized constructions, focus constructions, extraction and the bei constructions because the trace category and the filler are marked by co-indexation. For example, in Figure 6.1, the filler, marked as ‘NP-SBJ-3’, and the gap introduced in passive voice, marked as ‘NONE-*3’, are clearly annotated in the tree. With this information, we can know where to introduce -g category in GCG annotation and when to ‘fill’ the gap.

We also annotate various types of relative clause as nonlocal constructions. Although in Treebank annotations of relative clauses, as shown in Figure 6.2, the trace categories are marked to be co-indexed with an always empty operator ‘WHNP-1’, the head noun of the relative clause can be easily inferred from the syntactic structure. We did not include a type of relative clause where the head noun is an adverbial modifier of the relative clause, as shown in (45). In (45), the head noun 原因 ‘reason’ is annotated in the Penn Chinese Treebank as a moved adverbial modifier of the verb 赌博 ‘gambles’. We did not include this type of relative clause as a nonlocal construction because this type of relative clause has identical syntactic structure to an appositive clause in Chinese. For example, the difference between (45) and (46) is mostly semantic, rather than syntactic. It is hard to justify proposing a gap category in (45) but not in (46).

(45) 这 就 是 他 经 常 赌 博 的 原 因
    this exactly is he often gamble de reason
    ‘That’s the reason for his frequent gambling.’

(46) 这 就 是 他 经 常 赌 博 的 后 果
    this exactly is he often gamble de consequence
    ‘That’s the consequence for his frequent gambling.’
The constructions annotated for this experiment might not constitute an exhaustive list of the nonlocal constructions in the Penn Chinese Treebank. For example, trace categories are annotated for the verb 出现 ‘emerge’ in the sentence shown in Figure 6.18. However, we do not annotate nonlocal dependencies for this construction because the filler is not present linguistically. Since the parsing performance is evaluated per construction, adding or deleting some construction from the set of nonlocal constructions we plan to evaluate should not affect the parsing evaluation of other constructions in the set.

In total, we annotated eight types of nonlocal constructions, three types of relative clauses, two types of bei constructions, topicalization, focus construction and extraction from embedded clauses.

**Subject relative clause**

(47) 没有 影像 的 新闻 要 丢掉
do-not-have video de news needs dump
‘dump the news which do not have videos’

1(没有 ‘do-not-have’, 新闻 ‘news’)

Subject relative clause refers to the construction in which a subject is extracted from a relative clause and a trace category is annotated in the subject position in the relative clause. In Mandarin Chinese, a relative clause is followed by a particle de and the head noun occurs after the de particle. For example, in (47), the head noun 新闻 ‘news’ occurs after de and serves as the subject of the verb 没有 ‘do-not-have’ in the relative clause. We label the relation ‘1’ between the extracted subject, 新闻 ‘news’, and the verb, 没有 ‘do-not-have’, to indicate that the extracted subject is the first argument of the verb. We exclude those cases from our test sets where the head noun of the relative clause is not specified linguistically, so that a concrete dependency can be established between a pair of words.

**Object relative clause**

92
(48) 这些都不能带来他需要的幸福
these all cannot bring he needs de happiness
‘All these cannot bring the happiness that he needs’

2(需要 ‘need’, 幸福 ‘happiness’)

Object relative clause refers to the construction where the object is extracted from the relative clause as shown in (48). A label ‘2’ is given to the dependency between the extracted object, 幸福 ‘happiness’, and the verb, 需要 ‘need’. It means the extracted object is the direct object of the verb. We have not found any extracted indirect objects in relative clauses in the Treebank data.

Topic relative clause

(49) 胡子 长过 头发的维利阿
beard longer-than hair de Welia
‘Welia, whose beard is longer than his hair’

of-asso(胡子 ‘beard’, 维利阿 ‘Welia’)

Topic relative clause refers to the construction where the topic is extracted from the relative clause. In all the topic relative clauses we examined, we find there is no predicate-argument relation between the extracted topic and the verb in the relative clause. Instead, the extracted topic has a semantic relation with the subject of the relative clause which is similar to the relation in English expressed by ‘the subject of the topic’. For example, in (49), the extracted topic 维利阿 is not an argument of the verb 长过 ‘longer-than’. It is related to the subject 胡子 ‘beard’ instead. Therefore, we labeled a ‘of-asso’ dependency between the extracted topic and the subject of the relative clause.

Focus construction

(50) 他睡觉的地方都没有
he sleep de place even do-not-have
‘He does not even have a place to sleep.’

2(没有 ‘do-not-have’, 地方 ‘place’)

Focus constructions have the focused constituent occur preverbally. In (50), 睡觉的地方 ‘place to sleep’ is the focused constituent which occurs before the verb 没有 ‘do-not-have’, rather than after the verb, the canonical position of a direct object. Since 睡觉的地方 ‘place to sleep’ is the direct object of 没有 ‘do-not-have’, there is a labeled ‘2’ dependency between them.

Passivization of direct objects

(51)  公共 场合 抽烟 应该 被 禁止
public place smoking should be forbid
‘Smoking in public should be forbidden’

2(禁止 ‘forbid’, 抽烟 ‘smoking’)

Passivization of direct object refers to a passivized sentence where the direct object is fronted to become the subject of the sentence. Most passivized sentences in the Treebank data belong to this category. We give the label ‘2’ to the dependency between the verb and the subject since the subject is the second argument of the transitive verb.

Passivization of indirect objects

(52)  两人 被 给予 难民 身份
two-people bei given refugee status
‘Two people were given the status of refugee’

3(给予 ‘give’, 两人 ‘two-people’)

Passivization of indirect object refers to a passivized sentence where the indirect object is fronted to be the subject of the sentence. We give the label ‘3’ to the dependency between the ditransitive verb and the fronted object since it is the third argument of the verb.
Topicalization

(53) 这种事情你不是第一次碰到
this-sort-of things you not first-time came-across
‘It is not the first time that you came across this sort of things.’

2(碰到 ‘came-across’, 事情 ‘things’)

Topicalization refers to the construction where a word or phrase is moved to the sentence initial position to serve as a topic of the sentence. The dependency label could be ‘1’ or ‘2’, depending on whether the fronted word or phrase is the first argument or the second argument of the verb. In (53), the direct object 事情 ‘things’ is moved to the front of the sentence. Therefore a label ‘2’ is given to the dependency between 碰到 ‘came-across’ and 事情 ‘things’.

Extraction from an embedded clause

(54) 手机可以说是这个时代的一部分
cell-phone can say is this age de a part
‘Cell phone, we can say, is a part of this age’

1(是 ‘is’, 手机 ‘cell-phone’)

Extraction from an embedded clause refers to the construction where a subject or an object is moved across at least two clause boundaries. Many cases of extraction from an embedded clause we have examined actually belong to topicalization. We separate them out because we want to see whether extraction from embedded clauses is harder for an automatic parser. In (54), the noun phrase 手机 ‘cell-phone’ is extracted to the sentence initial position from its subject position within the complement clause of 说 ‘say’. A ‘1’ dependency is annotated between 是 ‘is’ and 手机 ‘cell phone’ to indicate that ‘cell-phone’ is the first argument of ‘is’.
6.2.3 The Data

We divided the Chinese Penn Treebank 6 into training, development and test sets as indicated in Tse and Curran (2010). From the test set and all the other sentences in the Penn Chinese Treebank 8 that are not included in the training set, for each nonlocal construction, we randomly choose 120 sentences to annotate with dependencies. If we could not find 120 cases for some rare constructions, we include all the occurrences from the test set in order to have a test set of reasonably large size. We annotate all the dependencies according to their trace annotations in the Penn Chinese Treebank. We removed the sentences where either the head or the dependent in the nonlocal dependency is not present linguistically. Sometimes, there is more than one occurrence of a particular construction in one sentence. In these cases, more than one nonlocal dependency is annotated. The size of the test set for each construction and its distribution in the Penn Chinese Treebank are shown in Table 6.2.

We can see subject relative clauses and object relative clauses are very common in the corpus. Around a third of sentences in the Penn Chinese Treebank contain at least a subject relative clause or an object relative clause. Focus constructions, topicalizations, passivization with fronted indirect
objects and extractions from an embedded clause occur relatively rarely in the corpus. It is possible that some constructions, such as the focus construction, can be more frequent in more colloquial text.

6.3 Experiments

We first examined the proportion of nonlocal dependencies annotated in each test set that can be recovered from some widely used dependency formalisms. Then we evaluated the performance of several automatic parsers on the task of recovering nonlocal dependencies to have a understanding of how difficult it is to recover each of these nonlocal dependencies.

6.3.1 Comparison between dependency formalisms

In this section, we examined some widely used dependency formalisms in terms of their coverage of the nonlocal dependencies we annotated in our test sets.

Stanford dependencies (Chang et al., 2009; de Marneffe et al., 2006) are probably the most widely used dependency formalism in recent years and can be easily obtained from Treebank annotations. Constituent parsers, such as the Stanford parser (Klein and Manning, 2003), the Berkeley parser (Petrov and Klein, 2007) and the Brown parser (Charniak, 2000), can all be trained on Treebank annotations to yield Stanford dependencies. Dependency parsers, such as MaltParser (Nivre et al., 2006), Mate (Bohnet, 2010) and MSTParser (MacDonald and Pereira, 2006), can be trained directly on Stanford dependencies to predict Stanford dependencies. If Stanford dependencies can reliably contain the nonlocal dependencies we annotated for Mandarin Chinese, we can easily evaluate a parser’s ability to recover nonlocal dependencies via Stanford dependencies.

Stanford dependencies come in a different format from the nonlocal dependencies we annotated. Stanford dependencies have more fine-grained dependency labels for some type of dependencies and more generic label for others. Here we follow the practice of Rimell et al. (2009) to convert Stanford
dependencies into the format of the dependencies we annotated. Heuristically, we mapped ‘nsubj’ in Stanford dependencies into ‘1’ dependencies in our annotations, ‘dobj’ into ‘2’ dependencies and ‘iobj’ into a ‘3’ dependencies. For passive voice, Stanford dependencies have a ‘nsubjpass’ dependency between the subject and the verb in a passivized sentence. In most of the cases, the subject is the second argument of the verb. Therefore, ‘nsubjpass’ dependencies are converted into ‘2’ dependencies in our annotations. This will not affect the results of passivization of direct objects but we will not be able to get results for passivization of indirect objects from Stanford dependencies.

For relative clauses, Chinese Stanford dependencies have a generic dependency labeled ‘rcmod’ between the head noun and the main verb of a relative clause, regardless of the type of the relative clause. For example, the Stanford dependencies of (47) will contain ‘rcmod(news, do-not-have)’ and the Stanford dependencies of (48) will contain ‘rcmod(happiness, need)’. However, this dependency label does not provide any information about whether the head noun is the first or second argument of the verb. After examining the statistics of relative clauses in Stanford dependencies of the training set, we found that when the subject is present in the relative clause while the object is missing, the relative clause is most likely to be an object relative clause; when the subject is missing but the object is present, it is most likely to be a subject relative clause; if both subject and object are absent, it is most likely to be a subject relative clause. By these principles, we mapped the rcmod dependencies into subject or object relative clause dependencies in our annotations. However, it is more difficult to recover nonlocal dependencies for topic relative clauses. For example, Stanford dependencies have ‘rcmod(Welia, longer-than)’ for (49), while ‘Welia’ is not a participant of ‘longer-than’. Although we can map all the relative clauses where both the subject and object are present into topic relative clauses, our statistics show that relative clauses with both subject and object present are most likely to be a relative clause relativizing an adverbial modifier as (45). Mapping the relative clause with
both subject and object present into topic relative clauses is not supported by our data. Therefore, we cannot compare the labeled dependency results of topic relative clauses from Stanford dependencies.

Another dependency formalism we hoped we could include in our comparison is the LTH dependencies (Johansson and Nugues, 2007). According to Choi et al. (2015), LTH dependencies include long-distance dependencies and the conversion from the constituent trees to the LTH dependency format uses the function tags of the tree structure so as to generate rich dependency structures including non-projective dependencies. However, the LTH dependencies do not include dependencies for the Chinese language. Instead, on its website, it recommended using Penn2Malt dependencies for Chinese. We therefore convert the gold Penn Chinese Treebank trees in the test sets into Penn2Malt dependencies and examine their coverage of the annotated nonlocal dependencies in our test sets. The Penn2Malt dependency comes in a similar format as the Stanford dependency. We applied similar heuristic conversion to the Penn2Malt dependencies to make them match the annotated nonlocal dependencies.

Considering the possibility that other dependency formalisms might give the dependency a different label that is not covered by our heuristic mapping, we also examine the coverage rate of unlabeled dependencies to check whether the pair of words which have a nonlocal dependency in our annotations are related in any type of dependency in these dependency formalisms. The coverage of nonlocal dependencies extracted from the gold Penn Chinese Treebank, either labeled (L) or unlabeled (U), are shown in Table 6.3.

GCG dependencies, extracted from the GCG annotations converted from the Treebank with full attention to the trace categories, have almost full coverage of the nonlocal dependencies annotated in the test set. In the very few cases where the GCG dependencies do not cover the annotated dependencies.

8http://nlp.cs.lth.se/software/treebank_converter/
<table>
<thead>
<tr>
<th>Construction</th>
<th>L/U</th>
<th>stf</th>
<th>p2m</th>
<th>gcg</th>
<th>gcg-a</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sbj rel</td>
<td>L</td>
<td>.97</td>
<td>.90</td>
<td>1.0</td>
<td>.62</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>.97</td>
<td>.90</td>
<td>1.0</td>
<td>.68</td>
</tr>
<tr>
<td>Obj rel</td>
<td>L</td>
<td>.63</td>
<td>.46</td>
<td>1.0</td>
<td>.63</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>.87</td>
<td>.73</td>
<td>1.0</td>
<td>.70</td>
</tr>
<tr>
<td>Tpc rel</td>
<td>L</td>
<td>-</td>
<td>-</td>
<td>1.0</td>
<td>.49</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>-</td>
<td>-</td>
<td>1.0</td>
<td>.53</td>
</tr>
<tr>
<td>Focus</td>
<td>L</td>
<td>.28</td>
<td>0.0</td>
<td>.99</td>
<td>.06</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>.97</td>
<td>.57</td>
<td>.99</td>
<td>.82</td>
</tr>
<tr>
<td>Pass_do</td>
<td>L</td>
<td>.42</td>
<td>.23</td>
<td>1.0</td>
<td>.83</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>.71</td>
<td>.51</td>
<td>1.0</td>
<td>.89</td>
</tr>
<tr>
<td>Pass_io</td>
<td>L</td>
<td>-</td>
<td>-</td>
<td>.96</td>
<td>.60</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>.69</td>
<td>.69</td>
<td>.96</td>
<td>.74</td>
</tr>
<tr>
<td>Topicalization</td>
<td>L</td>
<td>0.0</td>
<td>0.0</td>
<td>.98</td>
<td>.08</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>.86</td>
<td>.37</td>
<td>.98</td>
<td>.59</td>
</tr>
<tr>
<td>Embedded</td>
<td>L</td>
<td>.02</td>
<td>.02</td>
<td>1.0</td>
<td>.38</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>.04</td>
<td>.02</td>
<td>1.0</td>
<td>.42</td>
</tr>
</tbody>
</table>

Table 6.3: Nonlocal dependency coverage rates of Stanford dependency (stf), Penn2Malt dependency (p2m) and GCG dependencies converted from gold Penn Chinese Treebank trees and the coverage rate from an automatic parser trained on GCG annotations of Mandarin Chinese (gcg-a).
nonlocal dependencies, the nonlocal construction is embedded in an adverbial clause, which makes
the filler syntactically unreachable. (55) is one of these sentences.

(55) 日照市，被国家正式确定为
Rizhao，bei the-nation officially titled as
亚欧大陆东方桥头堡
the-new-oriental-bridgehead-of-Eurasian-continent
两年后，成为吸引
two-years-later, became attract
外资的热点城市。
foreign-investment de hot city

‘Rizhao, two years after it was titled by the nation as the new oriental bridgehead of the
Eurasian Continent, became a hot city attracting foreign investment’

For (55) a nonlocal dependency 2 (确定 ‘title’, 日照市 ‘Rizhao’) is annotated, because a trace
category is annotated for the object of 确定 ‘titled-’ in the Treebank annotations of (55) which is
coindexed with 日照市 ‘Rizhao’. However, both the Treebank annotations and the GCG derivation
of this sentence treat 被 国家 正式 确定 为 亚欧大陆东方桥头堡 两年后 ‘two years after it was
titled by the nation as the new bridgehead of the Eurasian Continent’ as an adverbial clause with a
pro-dropped subject. Therefore 日照市 ‘Rizhao’ becomes unreachable syntactically to be the filler
of the trace.

The Stanford dependencies do not have coverage on the labeled dependencies from topic
relative clause and passivation of indirect objects for the reasons we explained above. For other
constructions, it seems that the Stanford dependencies have high coverage for subject relative
clauses, but very low coverage for topicalization, extraction from an embedded clause and focus
construction.

Figure 6.20 and Figure 6.21 show two examples of the Stanford dependencies for the focus
construction and extraction from an embedded clause. The Treebank annotations in Figure 6.20 have
信心 ‘confidence’ coindexed with the object of 全失 ‘lose-completely’. Given this annotation, we
Figure 6.20: Stanford dependencies for 股市不断下跌投资人信心全失 ‘The stock market continues to fall and the investors completely lost their confidence’
Figure 6.21: Stanford dependencies for 星期三预料是克林顿外交之行的高潮 ‘Wednesday is expected to be the climax of Clinton’s diplomatic visit ’
expect an ‘nobj’ dependency, i.e. a verb-object relation, between 信‘confidence’ and 全失‘lose-completely’. However, we found the dependency dep(全失‘lose-completely’, 信心‘confidence’) instead. In the Stanford dependencies, ‘dep’ is a generic label for all those dependencies that the system could not pin down a specific label for. Given this dependency label, it is hard to recover the relation that 信心‘confidence’ is the second argument of 全失‘lose-completely’ as we annotated in the nonlocal dependency test sets.

For the sentence in Figure 6.21, we annotate 星期三‘Wednesday’ as the first argument of 是‘is’, according to the Treebank annotations. However, this pair of words, 是‘is’ and 星期三‘Wednesday’, are not related in any dependency provided by the Stanford dependencies.

The coverage from the Penn2Malt dependency is even worse than that of Stanford dependencies. In both Penn2Malt dependency and Stanford dependency, some nonlocal dependencies, even when unlabeled, are absent. In Table 6.3, ‘gcg-a’ shows the predicted results from a syntactic parser trained on the GCG annotations. We can see the automatically predicted dependencies can have better coverage than the Stanford dependencies or the Penn2Malt dependencies converted directly from the gold Treebank annotations for some nonlocal constructions, which indicates these two dependency formalisms cannot reliably be used to evaluate nonlocal dependency in Mandarin Chinese. The results also indicate that there is no need to compare any Stanford dependency based parsing systems in terms of nonlocal dependency recovery given the oracle coverage of the Stanford dependency is worse than the automatic parsing results from a parser trained on the GCG annotations.

### 6.3.2 Comparison between parsers

In this section, we compare the nonlocal dependency recovery performance from two different constituent parsers. We did not examine any dependency parsers because all the dependency parsers
we know are trained on projective dependencies where one word is associated with one head. However, in order to fully represent nonlocal dependency, the dependencies need to be represented in a graph where one node is allowed to connect with multiple other nodes. Therefore, we confined ourselves to constituent parsers, which can be trained on GCG derivations.

The two parsers we choose are the latent-variable PCFG Berkeley parser (Petrov and Klein, 2007) and the Berkeley self-attentive neural network parser (Kitaev and Klein, 2018). The Berkeley latent-variable PCFG parser has been among the top-performing parsers for its accuracy and speed for a decade. The Berkeley self-attentive neural parser uses a self-attentive architecture instead of an LSTM encoder to achieve new state-of-the-art parsing results for the Penn Treebank trees. It also achieves the best results in 8 out of the 9 languages in the SPMRL dataset. Therefore, we choose this parser as a representative to examine the performance of a neural network parser in terms of nonlocal dependency recovery.

The Berkeley latent-variable parser was trained with its default setting and five merge and splitting cycles, which is found to yield the best performance in our previous experiments. The neural network parser is also trained with the default hyper-parameters without using ELMO embeddings. Since the neural network parser needs part-of-speech tagged text for testing, we trained a Stanford log-linear part-of-speech tagger with the preterminals of the GCG derivations in the training set. When testing with the neural network parser, we passed the tagged test sets as the input.

We experimented with two different training sets, a small training set (gcg-s) and a large training set (gcg-l). The small training set is the same training set used in Tse and Curran (2010) which contains 15,957 sentences. The large training set includes all the sentences from the Penn Chinese Treebank 8 except those sentences used in nonlocal dependency test sets. The large training set contains 50,635 sentences. We trained two parsers with these two training sets and parsed the
<table>
<thead>
<tr>
<th>Construction</th>
<th>L/U</th>
<th>bkl-s</th>
<th>bkl-l</th>
<th>bnp-s</th>
<th>bnp-l</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sbj rel</td>
<td>L</td>
<td>.62</td>
<td>.64</td>
<td>.53</td>
<td>.64</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>.68</td>
<td>.74</td>
<td>.59</td>
<td>.71</td>
</tr>
<tr>
<td>Obj rel</td>
<td>L</td>
<td>.63</td>
<td>.77</td>
<td>.50</td>
<td>.60</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>.70</td>
<td>.81</td>
<td>.64</td>
<td>.70</td>
</tr>
<tr>
<td>Tpc rel</td>
<td>L</td>
<td>.49</td>
<td>.48</td>
<td>.26</td>
<td>.32</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>.53</td>
<td>.50</td>
<td>.28</td>
<td>.35</td>
</tr>
<tr>
<td>Focus</td>
<td>L</td>
<td>.06</td>
<td>.33</td>
<td>.02</td>
<td>.06</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>.82</td>
<td>.80</td>
<td>.61</td>
<td>.50</td>
</tr>
<tr>
<td>Pass_do</td>
<td>L</td>
<td>.76</td>
<td>.80</td>
<td>.69</td>
<td>.78</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>.83</td>
<td>.86</td>
<td>.74</td>
<td>.79</td>
</tr>
<tr>
<td>Pass_io</td>
<td>L</td>
<td>.60</td>
<td>.62</td>
<td>.23</td>
<td>.48</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>.72</td>
<td>.87</td>
<td>.68</td>
<td>.82</td>
</tr>
<tr>
<td>Topicalization</td>
<td>L</td>
<td>.08</td>
<td>.11</td>
<td>.01</td>
<td>.07</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>.59</td>
<td>.62</td>
<td>.38</td>
<td>.50</td>
</tr>
<tr>
<td>Embedded</td>
<td>L</td>
<td>.38</td>
<td>.31</td>
<td>.31</td>
<td>.15</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>.42</td>
<td>.33</td>
<td>.35</td>
<td>.17</td>
</tr>
</tbody>
</table>

Table 6.4: Nonlocal dependency coverage rates of Berkeley latent-variable parser (bkl) and the Berkeley self-attentive neural network parser (bnp) trained on a small (-s) and a large (-l) training set.

sentences in the nonlocal dependency test sets with these two parsers. Then we extracted the dependencies out of the parse outputs and evaluated the results against the dependencies annotated in the eight test sets.

In general, Table 6.4 shows that a larger training set is beneficial for recovering nonlocal dependencies for both the latent-variable parser and the neural network parser. Improvement has been observed in most constructions when the parser is trained with a larger training set. Small drops in accuracy are observed for topic relative clauses and extractions from an embedded clause.
<table>
<thead>
<tr>
<th>Parser</th>
<th>Standard test set</th>
<th>Gapeval test set</th>
</tr>
</thead>
<tbody>
<tr>
<td>latent-variable parser</td>
<td>82.62</td>
<td>76.70</td>
</tr>
<tr>
<td>Neural network parser</td>
<td>86.18</td>
<td>80.13</td>
</tr>
</tbody>
</table>

Table 6.5: Evalb F1 scores for the two parsers trained on the small training set

These usually only involve one or two more wrong predicted dependencies on the side of the large training set.

To our surprise, the latent-variable PCFG parser is better in recovering nonlocal dependencies than the neural network parser. The neural network parser is reported to have better overall parsing performance than that of the Berkeley latent-variable parser in English and eight other languages. We, therefore, use the widely used parsing evaluation algorithm, evalb,\(^9\) to evaluate the general parsing performance of these two parsers in predicting GCG derivations in Mandarin Chinese. We tested the two parsers on two test sets. The first test set, ‘standard test set’ in Table 6.5, is a test set of 2230 sentence used by Tse and Curran (2012). The second test set, ‘gapeval test set’ in Table 6.5, contains all the 736 sentences from the eight nonlocal construction test sets we annotated for the nonlocal dependency evaluation. The evalb algorithm compares the parse outputs of the two test sets from these two parsers with the gold GCG annotations and outputs the F1 scores of the comparison, as recorded in 6.5.

Results in Table 6.5 support the reported superior overall parsing performance of the neural network parser. On the ‘standard’ test set, the F1 score of neural network parser about 3.5% higher. The results also indicate that the ‘gapeval’ test set is harder to parse for both parsers, which is understandable because sentences containing nonlocal dependencies tend to be longer and more  

\(^9\)https://nlp.cs.nyu.edu/evalb/

107
syntactically complex. We also observe the difference in the parsing performance between these two parsers narrows on the ‘gapeval’ test sets.

In order to understand why a parser of overall better parsing performance does worse in nonlocal dependency recovery, we sampled a few sentences to inspect the mistakes the parser made. Figure 6.22 presents an example of passivization of an indirect object. Figure 6.22b shows that the parsing error of the neural network parser stems from the wrong preterminal for the verb 判处‘sentence’. In the few samples we examined, we found the mistaken parses from the neural network parser all involved some incorrectly predicted preterminals, which leads to a partially wrong parse of the sentence and the failure to recover the nonlocal dependencies. Since the preterminal tagger is trained separately, only taking into consideration the word sequence and previous tags, it is not optimized to maximize the likelihood of the whole tree structure. The tagger will be biased towards more frequent preterminals. A separate part-of-speech tagger might predict the Penn Treebank annotations, because the part-of-speech tags are relatively independent and fixed labels for words in the Treebank. However, in a categorial grammar framework, preterminals are a coherent part of the derivations. Also the number of categories of preterminals in a categorial grammar framework can be much larger than the number of part-of-speech tags, depending on how lexicalized the grammar is.

All these results seem to suggest that although the Berkeley neural network parser has better overall parsing performance, it is not necessarily better in recovering nonlocal dependencies, mainly because its separately trained preterminal tagger might not be accurate in predicting preterminals. The results also indicate that an overall syntactic parsing evaluation cannot replace the construction specific parsing evaluation, because the results of these two types of evaluation do not always correlate.
Figure 6.22: Predicted parses for 俩人被判处无期徒刑 ‘The two people was sentenced to life imprisonment’
6.4 Error analysis

In order to better understand the construction-specific parsing errors, we annotated a small development set for the eight nonlocal constructions. We did not have a development set for extractions from an embedded clause because of their rare occurrence in the data. The development set for each construction consists of seven to ten sentences. The error analysis below is conducted based on the parsing outputs of the Berkeley latent-variable parser on the development sets.

In spite of its frequent occurrence in the corpus, subject relative clauses are not easy to parse correctly as suggested by results in Table 6.3. Examining the failed cases in the development set suggests two potential difficulties for parsing subject relative clauses correctly. The first difficulty is that there are inconsistent Treebank annotations for the same noun phrase construction, as shown in Figure 6.23 and 6.24. There is no apparent semantic or syntactic reason to analyze the noun phrases in Figure 6.23 and 6.24 differently. This confusion caused the parser to be unable to reliably predict the internal structure of this type of noun phrases.

![Figure 6.23: Treebank annotations for 新的忧患 ‘new worries’](image)

Figure 6.23: Treebank annotations for 新的忧患 ‘new worries’

110
Another mistake observed in parsing subject relative clauses is caused by noun-verb confusion in Chinese.

(56) 傻冒 的 先生
  silly de man
  ‘a silly man’

傻冒‘silly’ in Chinese can either be a verb be silly or a noun a silly person. 傻冒‘silly’ is annotated as a verb in (56) in Treebank annotations and 傻冒的‘silly de’ forms a relative clause modifying 先生‘man’. However, the parser parses the word as a noun and the structure of the noun phrase is ‘noun + de + noun’, which is also a very common structure for noun phrases in Mandarin Chinese.

The errors in parsing object relative clauses are often caused by the difficulty of predicting correct internal structure of noun phrases. For example, a word sequence of ‘noun_1 + verb + de + noun_2 + de + noun_3’ is structurally ambiguous between two possible parses as shown in Figure 6.24: Treebank annotations for 新的特色 ‘new features’

Figure 6.24: Treebank annotations for 新的特色 ‘new features’
The difficulty of parsing topic relative clauses correctly is that there are multiple ways to parse the word sequence of a topic relative clause. A topic relative clause usually has a word sequence of ‘noun_1 + verb + (noun_2) + de + noun_3’ where noun_3 is the topic of the relative clause and we expect to have the dependency of-asso(noun_1, noun_3). The parser can parse noun_1 as adverbial modifier and the rest as a subject relative clause and yield the dependency 1(verb, noun_3). If the
noun_2 is not present, the parser is also likely to parse this word sequence as an object relative clause and yield the dependency 2(verb, noun_3). Since subject relative clauses and object relative clauses are both very frequent in the data, these two parsing possibilities are very competitive.

Table 6.3 shows that the parsing of a passivization with a fronted direct object is comparatively easy. This is because the passivization particle bei gives an unequivocal indication of the predicate-argument structure of the sentence. Since all ten sentences in the development set are parsed correctly, we speculate that the errors in the test set come from mistaken parsing of some noun phrase involved in the construction.

The parsing of passivizations with fronted indirect objects tends to be more challenging. Error analysis of the development set seems to suggest the parsing confusion between the second and third argument of a ditransitive verb accounts for most of the errors. In Chinese, the direct object and indirect object are passivized in the same way. For example, ’I was given a book’ is ‘I + bei + give + a book’ and ‘A book is given to me’ is ‘A book + bei + give + me’. Both of them are expressed with the word order ‘noun_1 + bei + verb + noun_2’. Therefore a parser can have a hard time deciding whether noun_1 is the second argument (direct object) or the third argument (indirect object) of the verb.

Focus constructions usually come with the word order ‘noun_1 + noun_2 + verb’ where noun_1 is the subject and noun_2 is the fronted object. However noun_1 is often not present linguistically because Chinese is a pro-drop language. In that case, noun_2 is often parsed mistakenly as the subject. Even when both noun_1 and noun_2 are both present, there is also a chance to parse either noun_1 or noun_2 as an adverbial modifier because of the relatively low frequency of focus constructions in the corpus.

Table 6.3 suggests that it is hard to correctly resolve the nonlocal dependencies in topicalizations and extractions from embedded clauses. The fact that these two constructions occur very rarely in
the data definitely contributes to the difficulty. Also, unlike relative clauses and passivized sentences, there is no syntactic marker to indicate the possible predicate-argument structure. The distance between the head and the dependent in these two constructions can be very long. An extra noun phrase at the beginning of a sentence can be parsed as a modifier or an adverbial rather than an extracted argument. Correctly resolving the nonlocal dependencies in these two constructions is very challenging.

6.5 Summary

In this section we first introduced the empty categories annotated in the Penn Chinese Treebank and the eight nonlocal constructions we annotated for the test sets. We examined the nonlocal dependency coverage in the gold Stanford dependencies, the Penn2Malt dependencies and the GCG dependencies. We found that the former two could not provide reliable coverage on the nonlocal dependencies we annotated in the test sets. We trained two parsers on the GCG annotations and evaluated the dependencies extracted from the predicted derivations from the two parsers. The results suggest that high scores in general syntactic parsing evaluation like evalb do not correlate with good performance in recovering nonlocal dependencies and resolving nonlocal dependencies can be a challenging task for some infrequent nonlocal constructions.
Chapter 7: Conclusion

This thesis describes a generalized categorial grammar which has a slightly larger set of inference rules and a much smaller set of categories compared to more strongly lexicalized categorial grammars. The slightly larger set of inference rules provides the flexibility to represent language-specific lexical and syntactic features in Mandarin Chinese effectively without introducing too many new categories into the grammar. The -g category in the grammar provides an intuitive account for the filler-gap constructions in Mandarin Chinese.

This thesis also provides a systematic way to convert the Penn Chinese Treebank annotations into this GCG formalism, making it possible to create a broad-coverage grammar for Mandarin Chinese in this formalism. The reannotation process takes full advantage of the trace information annotated the Treebank to make it possible to predict gap categories in the trees, which are essential in predicting nonlocal dependencies.

Our syntactic parsing evaluation results show that the Berkeley grammar trainer trained on the GCG annotations is significantly more accurate than the same parser trained on a more strongly lexicalized categorial grammar in predicting a common test set of binary unlabeled Penn Treebank trees. The much smaller set of categories in the GCG annotations leads to much lower lexical-categorial confusion rate and relieves the data sparsity problem for the parser trained on the GCG annotations.
In order to evaluate the Chinese GCG annotations without the constraints of a particular syntactic formalism, we built a Chinese semantic dependency parsing system based on the syntactic dependencies extracted from the predicted derivations of a parser trained on the Chinese GCG annotations. This system achieves comparable performance for the overall labeled dependency prediction and superior performance for the multi-headed dependency recovery in the Chinese semantic dependency parsing shared task of SemEval 2016.

Finally, we evaluated the Chinese GCG annotations on the coverage and prediction of construction specific nonlocal dependency in Mandarin Chinese. Since there are no Chinese nonlocal dependency test sets available for the evaluation, we annotated a set of tests sets for eight nonlocal constructions in Chinese based on their Treebank annotations. The Chinese GCG annotations show much better coverage of the nonlocal dependencies annotated in the test sets than other dependency formalism such as Chinese Stanford dependencies or the Penn2Malt dependencies in Chinese. Our parsing experiments show that some infrequent nonlocal constructions are very challenging for parsers and an overall higher parsing accuracy does not necessarily lead to better performance in recovering nonlocal dependency. Evaluating a parser against a construction-specific nonlocal dependency test sets helps to reveal some parsing challenges that regular bracket scoring matrices fail to. Also, evaluating a parser against nonlocal dependencies can help to promote better recovery of these dependencies, which can be semantically important for downstream applications such as question answering or event detection.

The fact that the nonlocal dependency recovery rate of an automatic parser trained on Chinese GCG annotations is higher than that of Chinese Stanford dependencies or Pen2Malt dependencies converted directly from gold Treebank trees shows that categorial grammar provides a better grammar formalism representing filler-gap phenomena. With gap information coded within the syntactic categories, a parser trained on categorial grammar annotations has the ability to predict
nonlocal dependencies without extra complexities. Our syntactic experiments show that the syntactic derivations in categorial grammar can be used to train an automatic parser as PCFG trees. The parser performance can improve as more sophisticated parsing algorithms are explored, which means that the gap information can be automatically predicted if a parser is trained using a categorial grammar formalism. The automatically predicted gap information can be used to resolve important natural language processing problems such as event extraction or question-answering. The fact that our semantic dependency parsing system which builds upon the features extracted from the GCG parser outputs achieved the best performance in predicting multi-headed dependencies verified the importance of gap information in resolving nonlocal dependencies. Although we currently have only the nonlocal dependency resolution results from our own parser, we believe a parser trained on annotations which has full access to gap information will be competitive in resolving nonlocal dependencies.
Bibliography


