Syntactic Function-Based Chinese Lexical Categories and Category Grammar Parsing

Qingjiang Wang[a], Lin Zhang[b], Chengguo Chang[a]
[a] School of Information Engineering, North China University of Water Resources and Electric Power, Zhengzhou 450011, China
Email: {wangqingjiang, changchengguo}@ncwu.edu.cn

[b] Modern Education Technology Center, Henan University of Economics and Law, Zhengzhou 450002, China
Email: zhanglin@huel.edu.cn

***Abstract*—By merging syntactic categories of word classes, lexical categories were obtained. By demonstrating combination and type raising rules respectively from curried and uncurried perspectives, a category combination algorithm was presented, in which application, composition and type raising rules were sequentially examined, and the first available rule was selected. A Chinese CCG parser was developed, including Chinese word segmentation, category annotation, and syntactic parsing, which could obtain all parsing trees for given category sequence, but only determinatively chose one to print. Experiments show the parser can correctly perform categorial derivations, and lexical categories determined by syntactic function are reasonable and acceptable.**

***Index Terms*—combinatory categorial grammar, lexical category, parser**

I. Introduction

Combinatory Categorial Grammar (CCG) [1][2] extends basic categorical grammar by adding rules for functional composition and type raising, making generative power mildly context-sensitive, and introduces slash modals to make combinatory rules cross-linguistic universal. CCG is fully lexicalized grammar formalism, and widely used for robust and large-scale natural language processing [3][4][5][6][7].

Lexical categories could be automatically extracted from CCG parsing Treebank [8][9][10]. Lexical category explicitly represents lexical syntactic function, while word classes are word clustering with same syntactic function, so theoretically lexical categories could also be manually determined by merging categories of word classes. Categorical ambiguity would be propagated through categorical dependency between word classes, which brings some complexity for transcendentally determining lexical categories according to syntactic knowledge, but in practice the lexical categories extracted from Treebank are also ambiguous, and categorial ambiguity is lexical inherent.

Algorithm CYK establishes chart from bottom to top by span increase, coinciding with category-combinatory bisectability, so naturally fits category grammar parsing. Analytic process could be divided into two stages. Firstly syntactic categories are assigned onto each word, then categorical combination is done according to categorial operating rules. Suppose the categorial number of lexical words are C_1, C_2, …, C_n, then the sequence space of lexical categories is $C_1 \times C_2 \times \ldots \times C_n$. The parser simply using algorithm CYK is ineffient, because it must try to establish one chart for every categorial sequence. By utilizing conditional probabilities of lexical categories to word contexts, C&C parser [11][12] initially assigns only a small number of categories to each word, then the parser attempts to find a spanning analysis using CYK algorithm. If one cannot be found, the parser requests more categories to build the chart again from scratch or repair the chart without rebuilding. The accuracy of conditional probabilities is high enough that the parser can find a spanning analysis using the initial category assignment in most cases. Ref. [13] developed a shift-reduce CCG parser using a discriminative model and beam search, which gives competitive accuracies compared to C&C.

Chinese word classification is done by word broad-sense conformation, while the change of word classes is decided by narrow-sense conformation, resultantly the relationship between word classes and syntactic constituents is not simple mapping. The categories of word classes are ambiguous and overlapped when they are determined by syntactic functions, lexical categories obtained by merging the categories of word classes also are ambiguous, which is not some drawback but the real reflection of lexical syntactic functions.

This paper describes the detailed steps for determining lexical categories, demonstrates operating properties of CCG rules, and proposes an algorithm for categorical combination, by which a Chinese CCG parser was implemented, including Chinese word segmentation, category annotation, syntactic parsing, and printing

Manuscript received September 23, 2013; revised October 5, 2013; accepted December, 2013. ©2005 IEEE.

This work was supported in part by the National Natural Science Foundation of China (GrantNo. 51304078), and the Project of Henan Province Education Departmental Science and Technology Research (GrantNo. 12B520003).

doi:10.4304/jsw.9.5.1270-1274

parsing trees. Finally, the correctness of lexical categories and categorial combination are evaluated by running the parser on some phrases or sentences.

II. Syntactic Function-based Lexical Categories

Word classes are word clustering with same syntactic function, namely serving as same set of syntactic constituents. Categories only can be assigned onto syntactic constituents, consequently each word class has multiple categories, namely categorial ambiguity, and word classes are discriminated by categorial lists.

Assume preliminary categories are $\{np, s\}$. If subject category is np, sentence category is s, then predicate category is $s\backslash_{*}np$. If predicate is verb-object structure, object category is np, then verb category is $s\backslash_{*}np/_{*}np$, and if verb has double objects, the verb category is $s\backslash_{*}np/_{*}np/_{*}np$. If the central constituent category is np, then modifier category is $np/_{*}np$, and complement category is $np\backslash_{*}np$. If central constituent is verb, its category is $s\backslash_{*}np/_{*}\$1$, where $\$1$ is any category, then adverbial modifier category is $s\backslash_{*}np/_{\diamond}(s\backslash_{*}np)$, and complement category is $s\backslash_{*}np\backslash_{\times}(s\backslash_{*}np)$. In similar way, the categories for other constituents can be obtained. According to syntactic constituents word classes can serve as, the categorial list of word classes can be determined. Empty words themselves do not act as syntactic constituents, but empty word phrases do, so the categories of empty words can be determined by phrase category and the categories of phrase-inside other components. Especially, conjunction category is $X/_{*}X\backslash_{*}X$ of $X\backslash_{*}X/_{*}X$, where X is any category. When determining word class categories, the slashes in categories select as low rule access privilege as possible, with the purpose of restricting the category combination capability. Some categories of word classes are listed in Table I, where sign | is category separator, modal $_{*}$ is suppressed.

TABLE I.
THE CATEGORIAL LISTS OF WORD CLASSES

Word class	Categorial list
n, nh	$np \mid np/np$
nt, nl	$np \mid np/np \mid s\backslash np/_{\diamond}(s\backslash np) \mid s/s \mid s\backslash np$
nd	$np \mid np/np \mid np\backslash np$
ns	$np \mid np/np \mid s\backslash np/_{\diamond}(s\backslash np) \mid s/s$
v	$s\backslash np \mid s\backslash np/np \mid s\backslash np/np/np$
vd	$s\backslash np\backslash_{\times}(s\backslash np)$
vu	$s\backslash np/_{\diamond}(s\backslash np)$
vl	$s\backslash np/np \mid s\backslash np/(np/np)$
a	$np/np \mid s\backslash np\backslash_{\times}(s\backslash np) \mid s\backslash np$
m	$np/np \mid np/_{\times}np \mid np\backslash np$
q	$np\backslash_{\times}np \mid np\backslash_{\diamond}np \mid s\backslash np\backslash_{\times}(s\backslash np)\backslash(np/np) \mid np/np$
r	$np \mid np/np \mid s\backslash np \mid s\backslash np/np \mid s\backslash np/_{\diamond}(s\backslash np)$
d	$s\backslash np/_{\diamond}(s\backslash np) \mid np/np/(np/np) \mid s/s$
p	$s\backslash np/_{\diamond}(s\backslash np)/np \mid s\backslash np\backslash x(s\backslash np)/np \mid np/np/np$
c	$X/X\backslash X$
u1(的)	$np/np\backslash np \mid np/np\backslash(np/np) \mid np/np\backslash(s\backslash np) \mid np/np\backslash(s/np) \mid np\backslash np \mid np\backslash(np/np) \mid np\backslash(s/np) \mid np\backslash(s\backslash np)$
u2(地)	$s\backslash np/_{\diamond}(s\backslash np)\backslash(np/np) \mid s\backslash np/_{\diamond}(s\backslash np)\backslash np$
u3(得)	$s\backslash np\backslash_{\times}(s\backslash np)/(np/np) \mid s\backslash np\backslash_{\times}(s\backslash np)/(s\backslash np\backslash_{\times}(s\backslash np)) \mid s\backslash np\backslash_{\times}(s\backslash np)/(s\backslash np) \mid s\backslash np\backslash_{\times}(s\backslash np)/s$
u4(着了过)	$s\backslash np\backslash_{\times}(s\backslash np)$

If a word is single-class word, then the categorial list of word class is just that of the word. If a word has multiple classes, then by merging categorial lists of these classes while reserving only one for same categories, the categorial list of the lexical word is obtained.

III. Operating properties of CCG rules

Suppose the slash priority is from left to right, namely categorial combination is left-first, so $A\backslash B/C=(A\backslash B)/C\neq A\backslash(B/C)$. The outmost bracket always can be removed.

Definition 3.1. (Category equivalence) The redundant brackets are removed from any two categories according to slash priority, the identical resultant category can be obtained, then the two categories are called equivalence.

Definition 3.2. (Category sameness) If the sign strings of any two categories are same with each other, then the two categories are called category sameness.

Definition 3.3. (Top slash and top subcategory) After removing redundant brackets according to slash priority, the slashes not belonging to any bracket are called top slashes, and the subcategories divided by top slashes are called top subcategories.

Definition 3.4. (Prefix category, host category, and prefix length) The sign string of category X is the prefix of sign string of category Y, then the X is prefix category of Y, Y is host category of X. If prefix category overlays left m top subcategories of host category, then prefix length is m.

In curried categories, prefix category with length n-1 is result category, and the nth top subcategory is argument category, here n is the number of top subcategories. In uncurried categories, the first top subcategory is result category, and the other top subcategories are argument categories. The slash to the left of argument category denotes argument directionality, slash / and \ means forward and backward respectively.

Usually CCGs consider the following eight rules, ① to ⑥ are combinatory rules, ⑦⑧ are type raising rules, and ③ to ⑥ also called composition rules. Slash subscripts denote modals, namely types, and arrow subscripts are rule names.

① Forward function application (>): $X/_{*}Y \quad Y \rightarrow_{>} X$

② Backward function application(<): $Y \quad X\backslash_{*}Y \rightarrow_{<} X$

③ Forward harmonic composition(<B): $X/_{\diamond}Y \quad Y/_{\diamond}Z \rightarrow_{>B} X/_{\diamond}Z$

④ Backward harmonic composition(<B): $Y\backslash_{\diamond}Z \quad X\backslash_{\diamond}Y \rightarrow_{<B} X\backslash_{\diamond}Z$

⑤ Forward crossed composition(>B×): $X/_{\times}Y \quad Y\backslash_{\times}Z \rightarrow_{>B\times} X\backslash_{\times}Z$

⑥ Backward crossed composition(<B×): $Y/_{\times}Z \quad X\backslash_{\times}Y \rightarrow_{<B\times} X/_{\times}Z$

⑦ Forward type raising(>T): $X \rightarrow_{>T} Y/.(Y\backslash.X)$

⑧ Backward type raising(<T): $X \rightarrow_{<T} Y\backslash.(Y/.X)$

Theorem 3.1. Without slash modals, if two categories can be combined, then there is only one combinatory rule available.

Proof: Considering the following combinatory rules, the annotations immediately after rules are categorical types from functional view. Rule ③ to ⑥ combine two function categories, primary and secondary functions are abbreviated respectively as priFunc and secFunc.

① X/Y Y →X	Function X/Y, argument Y
② Y X\Y →X	Function X\Y, argument Y
③ X/Y Y/Z →X/Z	PriFunc X/Y, secFunc Y/Z
④ Y\Z X\Y →X\Z	PriFunc X\Y, secFunc Y\Z
⑤ X/Y Y\Z →X\Z	PriFunc X/Y, secFunc Y\Z
⑥ Y/Z X\Y →X/Z	PriFunc X\Y, secFunc Y/Z

Combinatory rules imply categories are curried, so Y in X/Y or X\Y is just the rightmost top subcategory. For rule ① and ②, deciding whether argument category and Y in function category are equivalent is unambiguous. For ③ to ⑥, deciding whether the secondary categories include prefix Y is also unambiguous.

Rule ① and ② can be discriminated according to the slashes in function X/Y and X\Y. Rule ③ to ⑥ can be discriminated according to slash combinations after primary and secondary function are unique-two-categories divided successively. If the /Z and \Z are considered null, Rule ③ to ⑥ can be reduce to rule ① and ②. According to whether /Z and \Z are null, Rule ③ to ⑥ and rule ①② can be discriminated.

From the above, given two categories, Rule ① to ⑥ at most includes one rule available.

Theorem 3.2. Without type raising, if category A and B combine into category C, and given A and C, then B is uniquely determined.

Proof: Without loss of generality, assume A B→C. From Theorem 3.1, there exists only one available in rule ① to ⑥. If A is function category, only forward rules are potentially available. If rule ① is available, then A=C/B. Given A and C, B is uniquely determined. If rule ③ is available, then A=X/Y, B=Y/Z, and C=X/Z. If rule ⑤ is available, then A=X/Y, B=Y\Z, and C=X\Z. Under the two cases, A and C include the same prefix X, from which Y and Z are uniquely obtained, so B is uniquely determined. If B is function category, the proof can be obtained in same way.

From uncurried perspective, type raising rule raises argument category as function category, and the needed argument category is just the original function category or its prefix category. X↑ denotes the raised X, namely X↑=Y/(Y\X) or Y\(Y/X).

Theorem 3.3. If category X is the argument category of category Z, X↑ can combine with Z, then the argument category of X↑ is Z or prefix category of Z.

Proof: Assume X↑=Y/(Y\X), X↑ can forwardly combine with Z, then Z appears as Y\X, Y\X\$1, or Y\X/$1, where $1 is any category. Assume X↑=Y\(Y/X), X↑ can backwardly combine with Z, the similar analytical result can be obtained. [Proof end]

Seemingly, any category may be raised, but in practice the type-raising rule is used only when the argument category needs to be raised so that category combinatory order can change. If Z=Y\X, X as argument category can combine with Z using rule <, while the combinatory result does not change after X has been raised. If Z=Y\X\$1 or Y\X/$1, X may not combine with Z without raising, but can combine with Z using rule >B× or >B to obtain Y\$1 or Y/$1 respectively after raised. So only if X is argument category of Z and may not combine with Z, X is considered to be raised as X↑. Besides, type-raising rule may be used on demand, not only for raising lexical categories, but also for raising phrase categories, and thus enhances the acceptance capability of CCGs.

The slash modals form compatible hierarchy [8]. For modal $_{\star}$, only application rules are available. For modal $_{\diamond}$, application and harmonious composition rules are available. For modal $_{\times}$, application and crossed composition rules are available. For modal ., any rules are available. When using harmonious and crossed composition rules, the slash modal in resultant category should be same as that in secondary function. Slash modals restrict the rule selection scope, so Theorem 3.1, 3.2 and 3.3 are also true.

IV. Operating Properties-Based CCG Parser

The parser operation principle is illustrated as Fig. 1. Rule set includes rule ① to ⑧, and lexical categories database includes 41114 Chinese lexical entries with format word|category$_1$|category$_2$|…|category$_n$. Rule set together with lexical categories database composes the Chinese CCG. The average categorial number of lexical words is 2.80, which means lexical categories are ambiguous. Parsing trees represent the process of categorial combinations. If a word sequence has single corresponding category, CCG accepts the sequence, and print the parsing tree. If the category is s, CCG accepts the sequence as a sentence.

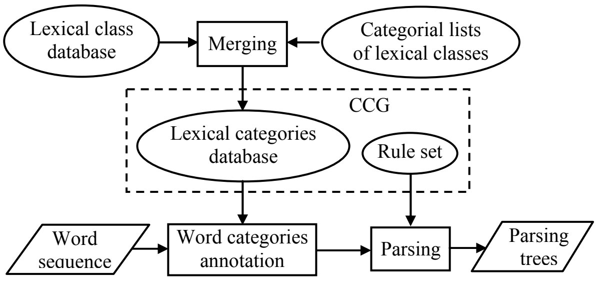
Lexical class database
Merging
Categorial lists of lexical classes
CCG
Lexical categories database
Rule set
Word sequence
Word categories annotation
Parsing
Parsing trees

Figure 1. The parser operation principle

The parsing algorithm uses CYK algorithm, and the basic operation is searching for the available rule to combine categories *cate1* and *cate2*. From Theorem 3.1, there at most is one available in rule ① to ⑥, while rule ⑦⑧ are considered only after no rule is available in rule ① to ⑥, so the rule availability may be examined by rule number order. The examining and computation of rule

①②③⑤⑦ are as follows, and those of rule ④⑥⑧ are in similar way.

① From curried perspective, If argument category of *cate1* is forward and equivalent to *cate2*, then result category of *cate1* is the combinatory result.

② From curried perspective, If argument category of *cate2* is backward and equivalent to *cate1*, then result category of *cate2* is the combinatory result. If *cate2* is just category $X/_{*}X\backslash_{*}X$，then *cate1*$/_{*}$*cate1* is the result.

③ From curried perspective, If argument category of *cate1* is equivalent to the prefix category with length *i* of *cate2*, and if the last top slash of *cate1* is $/_{\diamond}$ or /. , and the *i*th top slash of *cate2* starts with / not $/_{\times}$, then result category of *cate1* concatenates with the remains of *cate2* after removing the prefix category and *i*th top slash, getting the composition result.

⑤ From curried perspective, If argument category of *cate1* is equivalent to the prefix category with length *i* of *cate2*, and if the last top slash of *cate1* is $/_{\times}$ or /. , and the *i*th top slash of *cate2* starts with \ not $\backslash_{\diamond}$, then result category of *cate1* concatenates with the remains of *cate2* after removing the prefix category and *i*th top slash, getting the composition result.

⑦ From uncurried perspective, If the *i*th argument category of *cate2* is backward and equivalent to *cate1*, and the (*i*+1)th top slash of *cate2* starts with / not $/_{\times}$, or starts with \ not $\backslash_{\diamond}$, then the *i*th argument category is removed from *cate2*, and the remains is the composition result.

Usually each word has multiple selectable categories, and the whole word string has multiple category sequences. For each category sequence, the parsing algorithm tries all possible category combinations and creates parsing trees.

V. Experimental Results

From parsing trees, reasonability of lexical categories and correctness of categorial combinations can be decided. To eliminate spurious ambiguity, the derivations in CCGbank are in a formal form, which uses composition and type-raising only when syntactically necessary [14][15]. For simplicity, here the parser obtain all parsing trees for given category sequence, but only determinatively selects one to print, which is harmless for accompanying λ-term representation [2]. The parsing trees for following example sentences are in Fig. 2, 3 and 4, where sign $\diamond$ is replace with # to make programming easy, underlines and the signs on the right indicate combination and which rule has been applied. Apparently these trees are consistent with Chinese constituent parsing, and all category combinations are correct.

(1) 斯诺登住在机场
Snowden lives at the airport

(2) 达尔文在澳大利亚考察袋鼠
Darwin studied the kangaroo in Australia

(3) 达尔文提出的进化论改变了人类对世界的看法
The evolution theory Darwin proposed changed people’s view of the world

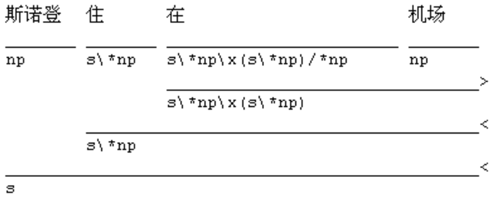

Figure 2. Parsing tree for example sentence (1)

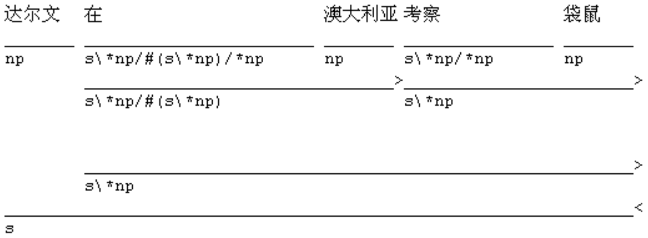

Figure 3. Parsing tree for example sentence (2)

Statistical results of processing different-length word sequences are in Table II, here *SeqLen* is word sequence length, *CateSeqNum*, *AcceptNum*, *AcceptAsSentNum* and *RunTime* are respectively the average of category sequence number, the accepted category sequence number, the accepted-as-sentence category sequence number, and parser running time. *Ambi* is average word category number, namely categorial ambiguity.

TABLE II.
The processing results of different word sequence lengths

SeqLen	*CateSeqNum*	*AcceptNum*	*AcceptAsSentNum*	*Ambi*	*RunTime*
4	48	2	1	2.63	896
5	856.8	9.2	3.4	3.86	1550
7	6912	4	2	3.54	3158
8	84000	37	22	3.67	24267
11	958464	156	96	3.44	388506

The parser attempts to establish parsing trees for every categorial sequence, so the running time rises quickly. For short sentences, only a few categorial sequences are accepted, among which the one most appropriate for Chinese constituent parsing always exists. The longer sentences are, the more categorial sequences are accepted, which means lexical categories still should be modified to restrict their combinatory capability.

VI. Conclusions

Combinatory Category Grammar has cross-linguistic operating rules and language-dependent lexical categories. Lexical categories can be extracted from CCG parsing Treebank, or manually determined by syntactic knowledge. Here proposes a syntactic function-based method to determine Chinese lexical categories. Category combination rules and type-raising rules are demonstrated respectively by curried and uncurried perspectives, and a category combination algorithm is presented, based on which a comprehensive Chinese CCG parser is developed. Experiments show the parser can correctly give parsing trees, and it is reasonable and feasible to determine

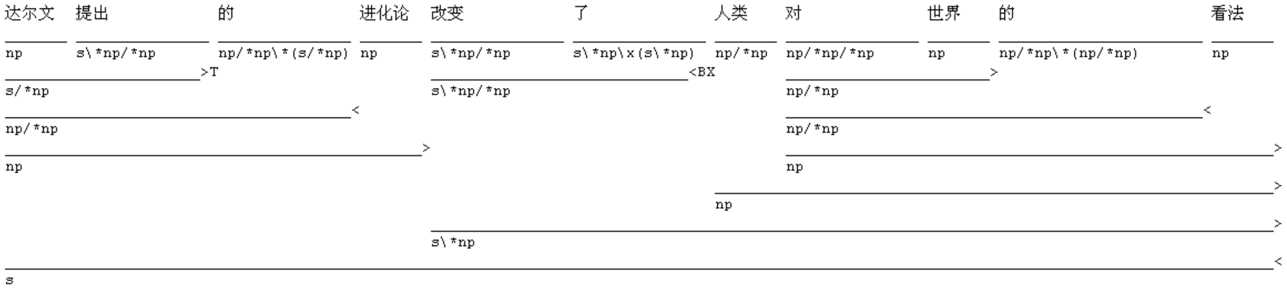
达尔文 提出 的 进化论 改变 了 人类 对 世界 的 看法
np s*np/*np np/*np*(s/*np) np s*np/*np s*np\x(s\np) np/np np/*np/*np np np/*np*(np/*np) np
>T
s/*np
<BX
s*np/*np
>
np/*np
<
np/*np
<
np/*np
>
np
>
np
>
np
>
s*np
<
s

Figure 4. Parsing tree for example sentence (3)

lexical categories by syntactic functions. Refining word classification could reduce category ambiguity so as to restrict combinatory capability, which will be done in the future.

REFERENCES

[1] S. Mark, and B. Jason, *Combinatory category grammar*. Blackwell: Non-Transformational Syntax, 2011, pp. 181-224.

[2] S. Mark, *Intuitive basis of combinatory categorial grammars*, http://ling.umd.edu//~alxndrw/CGReadings/steedman-31-40.pdf, 2013-8-6.

[3] S. Mark, B. Jason, and B. Çem, *Combinatory categorical grammars for robust natural language processing*. University of Edinburgh, 2012.

[4] C. Stephen, C. Ann, C. R. James, Z. Yue, H. Aurelie, H. James, et al, *Large-scale syntactic processing: Parsing the Web*. Johns Hopkins University, 2009.

[5] W. Fai, O. Francisco, and L. Yiping, "Hybrid machine aided translation system based on constraint synchronous grammar and translation corresponding tree," *Journal of Computers*, vol. 7, pp. 309-316, 2012.

[6] Z. Chun-Xiang, R. Ming-Yuan, L. Zhi-Mao, L. Ying-Hong, S. Da-Song, and L. Yong, "Multiple linear regression for extracting phrase translation pairs," *Journal of Computers*, vol. 6, pp. 905-912, 2011.

[7] Y. Ping-Fang, and D. Jia-Li, "Towards a syntactic structural analysis and an augmented transition explanation: A comparative study of the globally ambiguous sentences and Garden Path sentences," *Journal of Computers*, vol. 7, pp. 196-206, 2012.

[8] H. Julia, and S. Mark, "CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank," *Computational Linguististics*, vol. 33, pp. 355-396, 2007.

[9] B. Jason, *Lexically specified derivational control in combinatory categorical grammar*. University of Edinburgh, 2002.

[10] C. Stephen, and C. R. James, "Wide-coverage efficient statistical parsing with CCG and log-linear models," *Computational Linguistics*, vol. 33, pp. 493-552, 2007.

[11] C. Stephen, and C. R. James, "The importance of supertagging for wide-coverage CCG parsing," Proc 20th International Conference on Computational Linguistics, Geneva, Switzerland, pp. 282-288, 2004.

[12] D. Bojan, C. R. James, and C. Stephen, "Improving the efficiency of a wide-coverage CCG parser," Proc 10th International Conference on Parsing Technology, Prague, Czech Republic, pp. 39-47, 2007.

[13] Z. Yue, and C. Stephen, "Shift-reduce CCG parsing," Proc 49th Annual Meeting of the Association for Computational Linguistics, Portland, Oregon, pp. 683-692, 2011.

[14] M. Hepple, and G. Morrill, "Parsing and derivational equivalence," Proc 4th Conference of the European Chapter of the Association for Computational Linguistics, Manchester, UK, pp.10-18, 1989.

[15] J. Eisner, "Efficient normal-form parsing for Combinatory Categorial Grammar," Proc 34th Annual Meeting of the Association for Computational Linguistics, Santa Cruz, CA, pp. 79-86, 1996.

Qingjiang Wang received his B.E. degree in automatic control from Beijing University of Aeronautics and Astronautics, China, in 1990, and his M.E. and D.E. degrees in computer architecture from Xi'an Jiaotong University, China, in 2002 and 2005 respectively. He is an assistant professor at the School of Information Engineering, North China University of Water Resources and Electronic Power, Zhengzhou, China. His research interests include natural language processing and cloud computing.

Lin Zhang received her B.E. degree in computer application from Northwestern Polytechnical University, Xi'an, China, in 1991, and her M.E. degree in computer software from Information Engineering University of the People's Liberation Army, Zhengzhou, China, in 2005. She is an assistant professor at the Modern Education Technology Center, Henan University of Economics and Law, Zhengzhou, China. Her research interests include artificial intelligence, computational linguistics and cloud computing.

Chengguo Chang received her B.E. and M.E. degrees in computer science and technology from North China University of Water Resources and Electronic Power, Zhengzhou, China, in 2002 and 2005 respectively. She is a lecturer at the School of Information Engineering, North China University of Water Resources and Electronic Power. Her research interests include computational linguistics and data mining.