A Syntactic Resource for Thai: CG Treebank

Taneth Ruangrajitpakorn Kanokorn Trakultaweekoon Thepchai Supnithi
Human Language Technology Laboratory
National Electronics and Computer Technology Center
112 Thailand Science Park, Phahonyothin Road, Klong 1,
Klong Luang Pathumthani, 12120, Thailand
+66-2-564-6900 Ext.2547, Fax.: +66-2-564-6772
{taneth.ruangrajitpakorn, kanokorn.trakultaweekoon, thepchai.supnithi}@nectec.or.th

Abstract

This paper presents Thai syntactic resource: Thai CG treebank, a categorial approach of language resources. Since there are very few Thai syntactic resources, we designed to create treebank based on CG formalism. Thai corpus was parsed with existing CG syntactic dictionary and LALR parser. The correct parsed trees were collected as preliminary CG treebank. It consists of 50,346 trees from 27,239 utterances. Trees can be split into three grammatical types. There are 12,876 sentential trees, 13,728 noun phrasal trees, and 18,342 verb phrasal trees. There are 17,847 utterances that obtain one tree, and an average tree per an utterance is 1.85.

1 Introduction

Syntactic lexical resources such as POS tagged corpus and treebank play one of the important roles in NLP tools for instance machine translation (MT), automatic POS tagger, and statistical parser. Because of a load burden and lacking linguistic expertise to manually assign syntactic annotation to sentence, we are currently limited to a few syntactical resources. There are few researches (Satayamas and Kawtrakul, 2004) focused on developing system to build treebank. Unfortunately, there is no further report on the existing treebank in Thai so far. Especially for Thai, Thai belongs to analytic language which means grammatical information relying in a word rather than inflection (Richard, 1964). Function words represent grammatical information such as tense, aspect, modal, etc. Therefore, to recognise word order is a key to syntactic analysis for Thai. Categorial Grammar (CG) is a formalism which focuses on principle of syntactic behaviour. It can be applied to solve word order issues in Thai. To apply CG for machine learning and statistical based approach, CG treebank, is initially required.

CG is a based concept that can be applied to advance grammar such as Combinatory Categorial Grammar (CCG) (Steedman, 2000). Moreover, CCG is proved to be superior than POS for CCG tag consisting of fine grained lexical categories and its accuracy rate (Curran et al., 2006; Clark and Curran, 2007).

Nowadays, CG and CCG become popular in NLP researches. There are several researches using them as a main theoretical approach in Asia. For example, there is a research in China using CG with *Type Lifting* (Dowty, 1988) to find features interpretations of undefined words as syntactic-semantic analysis (Jiangsheng, 2000). In Japan, researchers also works on Japanese categorial grammar (JCG) which gives a foundation of semantic parsing of Japanese (Komatsu, 1999). Moreover, there is a research in Japan to improve CG for solving Japanese particle shifting phenomenon and using CG to focus on Japanese particle (Nishiguchi, 2008).

This paper is organised as follows. Section 2 reviews categorial grammar and its function. Section 3 explains resources for building Thai CG treebank. Section 4 describes experiment result. Section 5 discusses issues of Thai CG treebank. Last, Section 6 summarises paper and lists up future work.

Proceedings of the 7th Workshop on Asian Language Resources, ACL-IJCNLP 2009, pages 96–102,
Suntec, Singapore, 6-7 August 2009.

2 Categorial Grammar

Categorial grammar (Aka. CG or classical categorial grammar) (Ajdukiewicz, 1935; Carpenter, 1992; Buszkowski, 1998; Steedman, 2000) is a formalism in natural language syntax motivated by the principle of constitutionality and organised according to the syntactic elements. The syntactic elements are categorised in terms of their ability to combine with one another to form larger constituents as functions or according to a function-argument relationship. All syntactic categories in CG are distinguished by a syntactic category identifying them as one of the following two types:

1. Argument: this type is a basic category, such as s (sentence) and np (noun phrase).
2. Functor (or functor category): this category type is a combination of argument and operator(s) '/' and '\'. Functor is marked to a complex lexicon to assist argument to complete sentence such as s\np (intransitive verb) requires noun phrase from the left side to complete a sentence.

CG captures the same information by associating a functional type or category with all grammatical entities. The notation α/β is a rightward-combining functor over a domain of α into a range of β. The notation α\β is a leftward-combining functor over β into α. α and β are both argument syntactic categories (Hockenmaier and Steedman, 2002; Baldridge and Kruijff, 2003). The basic concept is to find the core of the combination and replace the grammatical modifier and complement with set of categories based on the same concept with fractions. For example, intransitive verb is needed to combine with a subject to complete a sentence therefore intransitive verb is written as s\np which means it needs a noun phrase from the left side to complete a sentence. If there is a noun phrase exists on the left side, the rule of fraction cancellation is applied as np*s\np = s. With CG, each lexicon can be annotated with its own syntactic category. However, a lexicon could have more than one syntactic category if it is able to be used in different appearances.

Furthermore, CG does not only construct a purely syntactic structure but also delivers a compositional interpretation. The identification of derivation with interpretation becomes an advantage over others.

Example of CG derivation of Thai sentence is illustrated in Figure 1.

Recently, there are many researches on combinatory categorial grammar (CCG) which is an improved version of CG. With the CG based concept and notation, it is possible to easily upgrade it to advance formalism. However, Thai syntax still remains unclear since there are several points on Thai grammar that are yet not completely researched and found absolute solvent (Ruangrajitpakorn et al., 2007). Therefore, CG is currently set for Thai to significantly reduce over generation rate of complex composition or ambiguate usage.

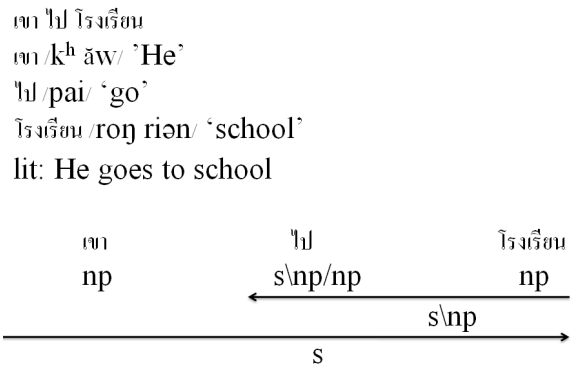

Figure 1. CG derivation tree of Thai sentence

3 Resources

To collect CG treebank, CG dictionary and parser are essentially required. Firstly, Thai corpus was parsed with the parser using CG dictionary as a syntactic resource. Then, the correct trees of each sentence were manually determined by linguists and collected together as treebank.

3.1 Thai CG Dictionary

Recently, we developed Thai CG dictionary to be a syntactic dictionary for several purposes since CG is new to Thai NLP. CG was adopted to our syntactic dictionary because of its focusing on lexicon's behaviour and its fine grained lexicalised grammar. CG is proper to nature of Thai language since Thai belongs to analytic language typology; that is, its syntax and meaning depend on the use of particles and word orders rather than inflection (Boonkwan, and Supnithi, 2008). Moreover, pronouns and other grammatical information, such as tenses, aspects, numbers, and voices, are expressed by function words such as

determiners, auxiliary verbs, adverbs and adjectives, which are in fix word order. With CG, it is possible to well capture Thai grammatical information. Currently we only aim to improve an accuracy of Thai syntax parsing since it still remains unresearched ambiguities in Thai syntax. A list of grammatical Thai word orders which are handled with CG is shown in Table 1.

Thai utilisation	Word-order
Sentence	- Subject + Verb + (Object)[1] [rigid order]
Compound noun	- Core noun + Attachment
Adjective modification	- Noun + Adjective[2]
Predicate Adjective	- Noun + Adjective[3]
Determiner	- Noun + (Classifier) + Determiner
Numeral expression	- Noun + (Modifier) + Number + Classifier + (Modifier)
Adverb modification	- Sentence + Adverb - Adverb + Sentence
Several auxiliary verbs	- Subject + (Aux verbs) + VP + (Aux verbs)
Negation	- Subject + Negator + VP - Subject + (Aux verb) + Negator + (Aux verb) + VP - Subject + VP + (Aux verb) + Negator + (Aux verb)
Passive	- Actee + Passive marker + (Actor) + Verb
Ditransitive	- Subject + Ditransitive verb + Direct object + Indirect object
Relative clause	- Noun + Relative marker + Clause
Compound sentence	- Sentence + Conjunction + Sentence - Conjunction + Sentence + Sentence
Complex sentence	- Sentence + Conjunction + Sentence - Conjunction + Sentence + Sentence
Subordinate clause that begins with word "ว่า"	- Subject + Verb + "ว่า"+ Sentence

Table 1. Thai word orders that CG can solve

In addition, there are many multi-sense words in Thai. These words have the same surface form but they have different meanings and different usages. This issue can be solved with CG formalism. The different usages are separated because the annotation of syntactic information. For example, Thai word "เกาะ" /kɔʔ/ can be used to refer to noun as an '*island'* and it is marked as **np**, and this word can also be denoted an action which means '*to clink*' or 'to attach' and it is marked as **s\np/np**.

After observation Thai word usage, the list of CG was created according to CG theory explained in Section 2.

Thai argument syntactic categories were initially created. For Thai language, six argument syntactic categories were determined. Thai CG arguments are listed with definition and examples in Table 2. Additionally, **np**, **num**, and **spnum** are a Thai CG arguments that can directly tag to a word, but other can not and they can only be used as a combination for other argument.

With the arguments, other type of word are created as functor by combining the arguments together following its behaviour and environmental requirements. The first argument in a functor is a result of combination. There are only two main operators in CG which are slash '/' and backslash '\' before an argument. A slash '/' refers to argument requirement from the right, and a backslash '\' refers to argument requirement from the left. For instance, a transitive verb requires one **np** from the left and one **np** from the right to complete a sentence. Therefore, it can be written as **s\np/np** in CG form. However, several Thai words have many functions even it has the same word sense. For example, Thai word "เชื่อ" /cʰɯ̂ə/ (to believe) is capable to use as intransitive verb, transitive verb, and verb that can be followed with subordinate clause. This word therefore has three different syntactic categories. Currently, there are 72 functors for Thai.

With an argument and a functor, each word in the word list is annotated with CG. This information is sufficient for parser to analyse an input sentence into a grammatical tree. In conclusion, CG dictionary presently contains 42,564 lexical entries with 75 CG syntactic categories. All Thai CG categories are shown in Appendix A.

[1] Information in parentheses is able to be omitted.

[2] Adjective modification is a form of an adjective performs as a modifier to a noun, and they combine as a noun phrase.

[3] Predicate adjective is a form of an adjective acts as a predicate of a sentence.

Thai argument category	definition	example
np	a noun phrase	ช้าง (elephant), ผม (I, me)
num	A both digit and word cardinal number	หนึ่ง (one), 2 (two)
spnum	a number which is succeeding to classifier instead of proceeding classifier like ordinary number	นึง (one), เดียว (one)[4]
pp	a prepositional phrase	ในรถ (in car), บนโต๊ะ (on table)
s	a sentence	ช้างกินกล้วย (elephant eats banana)
ws	a specific category for Thai which is assigned to a sentence that begins with Thai word ว่า (that : sub-ordinate clause marker).	* ว่าเขาจะมาสาย [5] 'that he will come late'

Table 2. List of Thai CG arguments

3.2 Parser

Our implemented lookahead LR parser (LALR) (Aho and Johnson, 1974; Knuth, 1965) was used as a tool to syntactically parse input from corpus. For our LALR parser, a grammar rule is not manually determined, but it is automatically produced by a any given syntactic notations aligned with lexicons in a dictionary therefore this LALR parser has a coverage including a CG formalism parsing. Furthermore, our LALR parser has potential to parse a tree from sentence, noun phrase and verb phrase. However, the parser does not only return the best first tree, but also all parsable trees to gather all ambiguous trees since Thai language tends to be ambiguous because of lacking explicit sentence and word boundary.

3.3 Tree Visualiser

To reduce load burden of linguist to seek for the correct tree among all outputs, we developed a tree visualiser. This tool was developed by using an open source library provided by NLTK: The Natural Language Toolkit (http://www.nltk.org/Home; Bird and Loper, 2004).

A tree visualiser is a tool to transform a textual tree structure to graphic tree. This tool reads a tree marking with parentheses form and transmutes it into graphic. This tool can transform all output types of tree including sentence tree, noun phrase tree, and verb phrase tree. For example, Thai sentence "|การ|ล่า|เสือ|เป็น|การ|ผจญภัย|" /ka:n lâ: sɯ̌ə pɤn ka:n pʰaʔ con pʰai/ 'lit: Tiger hunting is an adventure' was parsed to a tree shown in Figure 2. With a tree visualiser, the tree in Figure 2 was transformed to a graphic tree illustrated in Figure 3.

4 Experiment Result

In the preliminary experiment, 27,239 Thai utterances with a mix of sentences and phrases from a general domain corpus are tested. The input was word-segmented by JwordSeg (http://www.su-parsit.com/nlp-tools) and approved by linguists. In the test corpus, the longest utterance contains seventeen words, and the shortest utterance contains two words.

```
s
 (np
  (np/(s\np)[การ]
   s\np(
    (s\np)/np[ล่า]
    np[เสือ]
   )
  )
  s\np(
   (s\np)/np[เป็น]
   np(
    np/(s\np)[การ]
    s\np[ผจญภัย]
   )
  )
 )
```

Figure 2. An example of CG tree output

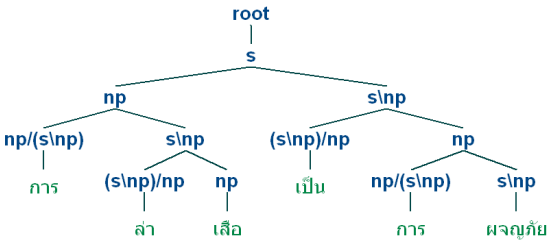

Figure 3. An example of graphic tree

[4] This spnum category has a different usage from other numerical use, e.g. ม้า[noun,'horse'] ตัว[classifier] เดียว[spnum,'one'] 'lit: one horse'. This case is different from normal numerical usage, e.g. ม้า[noun,'horse'] หนึ่ง [num,'one'] ตัว[classifier] 'lit: one horse'

[5] This example is a part of a sentence ฉันเชื่อว่าเขาจะมาสาย 'lit: I believe that he will come late'

All trees are manually observed by linguists to evaluate accuracy of the parser. The criteria of accuracy are:

- A tree is correct if sentence is successfully parsed and syntactically correct according to Thai grammar.
- In case of syntactic ambiguity such as a usage of preposition or phrase and sentence ambiguity, any tree following those ambiguity is acceptable and counted as correct.

The parser returns 50,346 trees from 27,239 utterances as 1.85 trees per input in average. There are 17,874 utterances that returns one tree. The outputs can be divided into three different output types: 12,876 sentential trees, 13,728 noun phrasal trees, and 18,342 verb phrasal trees.

From the parser output, tree amount collecting in the CG tree bank in details is shown in Table 3.

Tree type	Utterance amount	Tree amount	Average
Only S	8,184	12,798	1.56
Only NP	7,211	12,407	1.72
Only VP	8,006	11,339	1.42
Both NP and S	1,583	5,188	3.28
Both VP and S	1,725	6,816	3.95
Both NP and VP	397	1,140	2.87
S, NP, VP	133	658	4.95
Total	27,239	50,346	1.85

Table 3. Amount of tree categorised by a different kind of grammatical tree

5 Discussion

After observation of our result, we found two main issues.

First, some Thai inputs were parsed into several correct outputs due to ambiguity of an input. The use of an adjective can be parsed to both noun phrase and sentence since Thai adjective can be used either a noun modifier or predicate. For example, Thai input "|เด็กๆ|สดใส|บน|สนาม|" / dɛk dɛk sòd săi bon saʔ nă:m/ can be literally translated as follows:

1. Children is cheerful on a playground.
2. Cheerful children on a playground

For this problem, we decided to keep both trees in our treebank since they are both grammatically correct.

Second, the next issue is a variety of syntactic usages of Thai word. It is the fact that Thai has a narrow range of word's surface but a lot of polysymy words. The more the word in Thai is generally used, the more utilisation of word becomes varieties. With the several combination, there are more chances to generate trees in a wrong conceptual meaning even they form a correct syntactic word order. For example, Thai noun phrase "กำลัง|มหาศาล" /kam laŋ maʔ hă: să:n/ 'lit: great power' can automatically be parsed to three trees for a sentence, a noun phrase, and a verb phrase because of polysymy of the first word. The first word "**กำลัง**" has two syntactic usages as a noun which conceptually refers to *power* and a pre-auxiliary verb to imply progressive aspect. The word "**มหาศาล**" is an adjective which can perform two options in Thai as noun modifier and predicate. These affect parser to result three trees as follows:

np: np(np[กำลัง] np\np[มหาศาล])
s: s(np[กำลัง] s\np[มหาศาล])
vp: s\np((s\np)/(s\np)[กำลัง] s\np[มหาศาล])

Even though all trees are syntactically correct, only noun phrasal tree is fully acceptable in terms of semantic sense as *great power*. The other trees are awkward and out of certain meaning in Thai. Therefore, the only noun phrase tree is collected into our CG treebank for such case.

6 Conclusion and Future Work

This paper presents Thai CG treebank which is a language resource for developing Thai NLP application. This treebank consists of 50,346 syntactic trees from 27,239 utterances with CG tag and composition. Trees can be split into three grammatical types. There are 12,876 sentential trees, 13,728 noun phrasal trees, and 18,342 verb phrasal trees. There are 17,847 utterances that obtain one tree, and an average tree per an utterance is 1.85.

In the future, we plan to improve Thai CG treebank to Thai CCG treebank. We also plan to reduce a variety of trees by extending semantic feature into CG. We will improve our LALR parser to be GLR and PGLR parser respectively to reduce a missing word and named entity problem. Moreover, we will develop parallel Thai-English treebank by adding a parallel English treebank aligned with Thai since parallel treebank is useful resource for learning to statistical

machine translation. Furthermore, we will apply obtained CG treebank for automatic CG tagging development.

Reference

Alfred V. Aho, and Stephen C. Johnson. 1974 LR Parsing, In *Proceedings of Computing Surveys,* Vol. 6, No. 2.

Bob Carpenter. 1992. "Categorial Grammars, Lexical Rules,and the English Predicative", In R. Levine, ed., Formal Grammar: Theory and Implementation. OUP.

David Dowty, Type raising, functional composition, and non-constituent conjunction, In Richard Oehrle et al., ed., Categorial Grammars and Natural Language Structures. D. Reidel, 1988.

Donald E. Knuth. 1965. *On the translation of languages from left to right*, Information and Control 86.

Hisashi Komatsu. 1999. "Japanese Categorial Grammar Based on Term and Sentence". In *Proceeding of The 13th Pacific Asia Conference on Language, Information and Computation,* Taiwan.

James R. Curran, Stephen Clark, and David Vadas. 2006. Multi-Tagging for Lexicalized-Grammar Parsing. In *Proceedings of the Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (ACL)*, Paris, France.

Jason Baldridge, and Geert-Jan. M. Kruijff. 2003. "Multimodal combinatory categorial grammar". In *Proceeding of 10th Conference of the European Chapter of the ACL-2003,* Budapest, Hungary.

Julia Hockenmaier, and Mark Steedman. 2002. "Acquiring Compact Lexicalized Grammars from a Cleaner Treebank". In *Proceeding of 3rd International Conference on Language Resources and Evaluation (LREC-2002),* Las Palmas, Spain.

JWordSeg, word-segmentation toolkit. Available from: http://www.suparsit.com/nlp-tools), 2007.

Kazimierz Ajdukiewicz. 1935. *Die Syntaktische Konexitat*, Polish Logic.

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

NLTK: The Natural Language Toolkit. Available from: http://www.nltk.org/Home

Noss B. Richard. 1964. *Thai Reference Grammar*, U. S. Government Printing Office, Washington DC.

Prachya Boonkwan, and Thepchai Supnithi. 2008. Memory-inductive categorial grammar: An approach to gap resolution in analytic-language translation. In *Proceeding of 3rd International Joint Conference on Natural Language Processing (IJCNLP-2008)*, Hyderabad, India.

Stephen Clark and James R. Curran. 2007. Formalism-Independent Parser Evaluation with CCG and DepBank. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, Prague, Czech Republic.

Steven G. Bird, and Edward Loper. 2004. NLTK: The Natural Language Toolkit, In *Proceedings of 42nd Meeting of the Association for Computational Linguistics (Demonstration Track)*, Barcelona, Spain.

Sumiyo Nishiguchi. 2008. Continuation-based CCG of Japanese Quantifiers. In *Proceeding of 6th ICCS,* The Korean Society of Cognitive Science, Seoul, South Korea.

Taneth Ruangrajitpakorn, Wasan. na Chai, Prachya Boonkwan, Montika Boriboon, and Thepchai. Supnithi. 2007. The Design of Lexical Information for Thai to English MT, In *Proceeding of SNLP 2007*, Pattaya, Thailand.

Vee Satayamas, and Asanee Kawtrakul. 2004. Wide-Coverage Grammar Extraction from Thai Treebank. In *Proceedings of Papillon 2004 Workshops on Multilingual Lexical Databases,* Grenoble, France.

Wojciech Buszkowski, Witold Marciszewski, and Johan van Benthem, ed., *Categorial Grammar*, John Benjamin, Amsterdam, 1998.

Yu Jiangsheng. 2000. *Categorial Grammar based on Feature Structures*, dissersion in In-stitute of Computational Linguistics, Peking University.

Appendix A

Type	CG Category
Conjoiner	ws/s
Conjoiner	ws/(s/np)
Function word	spnum
Particle, Adverb	s\s
Verb	s\np/(s\np)/np
Verb	s\np
Function word, Particle	s/s
Function word	s/np
Auxiliary verb	s/(s/np)
Sentence	s
Conjoiner	pp/s
Conjoiner	pp/np
Conjoiner	pp/(s\np)
Function word	num
Classifier	np\num
Adjective	np\np
Noun, Pronoun	np/pp
Adjective, Determiner	np/np
Function word	np/(s\np)
Auxiliary verb	np/(np/np)
Function word	np/((s\np)/np)
Noun, Pronoun	np
Conjunction	(s\s)/s
Adverb, Auxiliary verb	(s\np)\(s\np)

Type	CG Category
Verb	(s\np)/ws
Verb, Adjective	(s\np)/pp
Determiner	(s\np)/num
Verb, Adjective	(s\np)/np
Function word, Verb, Adverb, Auxiliary verb	(s\np)/(s\np)
Function word	(s\np)/(np\np)
Auxiliary verb	(s\np)/((s\np)/np)
Conjunction	(s/s)/s
Function word	(s/s)/np
Function word	(s/s)/(s/np)
Classifier	(np\np)\num
Function word, Adverb, Auxiliary verb	(np\np)\(np\np)
Classifier	(np\np)/spnum
Function word	(np\np)/s
Determiner	(np\np)/num
Adjective, Conjoiner	(np\np)/np
Function word	(np\np)/(s\np)
Classifier, Function word, Adverb, Auxiliary verb	(np\np)/(np\np)
Auxiliary verb	(np\np)/((np\np)/np)
Adjective, Determiner	(np/pp)\(np/pp)
Determiner	(np/pp)/(np/pp)
Classifier	((s\np)\(s\np))\num
Classifier	((s\np)\(s\np))/spnum
Function word	((s\np)\(s\np))/np
Conjoiner	((s\np)\(s\np))/(s\np)

Type	CG Category
Function word	((s\np)\(s\np))/(np\np)
Function word	((s\np)\(s\np))/((s\np)\(s\np))
Verb	((s\np)/ws)/pp
Verb	((s\np)/ws)/np
Adverb, Auxiliary verb	((s\np)/pp)\((s\np)/pp)
Verb	((s\np)/pp)/np
Function word, Adverb	((s\np)/pp)/((s\np)/pp)
Auxiliary verb	((s\np)/np)\((s\np)/np)
Verb	((s\np)/np)/np
Verb	((s\np)/np)/(s\np)
Adverberb	((s\np)/(s\np))\((s\np)/(s\np))
Function word	((np\np)\(np\np))/np
Conjoiner	((np\np)\(np\np))/(np\np)
Adverb, Auxiliary verb	((np\np)/pp)\((np\np)/pp)
Adverb, Function word	((np\np)/pp)/((np\np)/pp)
Auxiliary verb	((np\np)/np)\((np\np)/np)
Conjoiner	((np/pp)\(np/pp))/(np/pp)
Verb	(((s\np)\np)
Verb	(((s\np)/ws)/pp)/np
Conjoiner	(((s\np)/pp)\((s\np)/pp))/((s\np)/pp)
Function word	(((s\np)/pp)\((s\np)/pp))/(((s\np)/pp)\((s\np)/pp))
Verb	(((s\np)/pp)/np)/np
Function word	(((s\np)/pp)/np)/(((s\np)/pp)/np)
Verb	(((s\np)/np)/(s\np))/pp
Conjoiner	(((np\np)/pp)\((np\np)/pp))/((np\np)/pp)