Korean Parsing in an Extended Categorial Grammar

Juyeon Kang

NLP Centre, Faculty of Informatics,
Masaryk University, Botanická 68a, 602 00 Brno, Czech Republic
qkang@aurora.fi.muni.cz

Abstract. This paper gives an automatic morpho-syntactical analysis with the ACCG parser which use the Categorial Grammar, the Combinatory Logic in the framework of Cognitive and Applicative Grammar. We focus on the contribution of the parser to the analysis of morphological case system in Korean.

Key words: Categorial Grammar, Combinatory Logic, Cognitive and Applicative Grammar, Korean Parsing, Case system

1 Introduction

The theoretical point of view of this work is that all linguistic units of Natural Languages are operators and operands with functional types. We show how it is possible to build up formal semantic representations from morpho-syntactical configurations using Extended Categorial Grammar and the combinators of Combinatory Logic (CL) with functional types [4] (Curry and al., 1972). CL is a useful formalism for studying the grammatical and lexical meanings [8] (Desclés, 1999). All expressions of CL are applicative expressions where an operator is applied to an operand. CL is generated from abstract operators, called combinators, whose aim is to combine more elementary operators (for instance, linguistic units into complex operators).

We give an automatic syntactical analysis with the ACCG parser by focusing on the analysis of morphological case system in Korean within the framework of the Cognitive and Applicative Grammar (CAG) model.

2 Frameworks

Categorial Grammars [7] are systems of types (analogue to Church's functional types); the instances of types are linguistic units analyzed as operators and operands. The calculus on syntactical types (or Lambek calculus, van Benthem, 1988) is closely associated to applications of operators onto operands. It has already been studied how Combinatory Logic [4] can be used with success for a semantic and computational analysis of voices, for instance in accusative and ergative natural languages [8]. The linguistic units being operators with assigned types, they can be composed by different ways. Thus CL is an adequate and "natural" formalism to express applicative expressions built by the

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2011, pp. 43–49, 2011.

application of operators to operands, and different compositions between operators. Indeed, CL is a logic of operators with abstract operators, called "combinators", which are used to compose and to define complex operators from more elementary operators. In an applicative calculus, combinators are introduced or eliminated by rules in Gentzen's style (Fitch, 1974). CL analyzes new concepts introduced in theories (logical, mathematical, linguistic, computer science, biology and nano-structure... theories) by an equivalence law between a definiendum and a definiens. The definiendum is a new unit and the definiens is an applicative expression where a combinator X describes how different more elementary operators are combined together.

The explicit articulation between morpho-syntactic configurations (organized by concatenation) and semantic and cognitive representations is described inside a computational architecture with intermediary levels. This architecture is defined in the formal and linguistic model of Cognitive and Applicative Grammar [1].

2.1 Cognitive and Applicative Grammar (CAG)

CAG is analogue to a compiling program with 7 interrelated levels of representations. This model, presented as a "bottom-up" analysis, which allows relating semantic representations and linguistic observables by means of formal calculus abstract.

The 7 levels are:

1. ***morpho-syntactical configuration level*** where the sentences are presented as concatenational strings (level (1));
2. ***operator-operand level*** is the result of an Extended Categorial Grammar analysis (ACCG parser); it is a set of applicative expressions associated to sentences of the level (1) (level (2));
3. ***analysis of diathesis and topicalisations*** in using combinators of Combinatory Logic (level (3)) [8];
4. ***analysis and representation of speaking acts*** for describing tenses, aspects (see below analyses), modalities and commitment operations (level (4));
5. ***formal representation of the meaning of lexical predicates by Semantic-Cognitive Schemes*** (SCS) (level (5));
6. ***integration of speaking conditions with SCS*** (level (6));
7. ***cognitive representation level*** (by diagrams or iconic representations) in relation with cognitive abilities of perception and action (level (7)).

This work concerns the levels (1), (2) and (3).

2.2 Applicative Combinatory Categorial Grammar

The Applicative Combinatory Categorial Grammar formalism is an extension of the Combinatory Categorial Grammar developed by Steedman. This ACCG formalism was originally developed by J-P. Desclés and I. Biskri (1995, 1996) for the analysis of coordination and subordination structure in French with the tools of Combinatory Logic by introducing canonical associations between some rules and the combinators.

We present here the rules[1] of the ACCG for the analysis of Korean sentences.

Application rules	
[X/Y : u_1]-[Y : u_2] ----------------------> [X : (u_1 u_2)]	[Y : u_1]-[X\Y : u_2] ----------------------< [X : (u_2 u_1)]
Type raising rules	
[X : u] ---------------------->**T** [Y/(Y\X) : (**C*** u)]	[X : u] ----------------------<**T** [Y\(Y/X) : (**C*** u)]
Functional composition rules	
[X/Y : u_1]-[Y/Z : u_2] ------------------------->**B** [X/Z : (**B** u_1 u_2)]	[Y\Z : u_1]-[X\Y : u_2] -------------------------<**B** [X\Z : (**B** u_2 u_1)]

Fig. 1. ACCG's rules

To the two classical basic types N(nominal) and S(sentence), we add a new basic type N* for the complete nominal phrases.

We use predefined notations to facilitate our categorial analysis.

$$X^0=S$$
$$X^1=(S\backslash N^*)$$
$$X^2=(S\backslash N^*)\backslash N^*$$
$$X^3=((S\backslash N^*)\backslash N^*)\backslash N^*$$

[1] **B** is a composition combinator. Its β-reduction is: Bfgx->f(gx). It is joined to the functional composition rule. This combinator allows us in particular to handle the free word order structure in the Korean sentence. **C*** is a type raising combinator joined to the type raising rule. Its β-reduction is: C*fg->gf This combinator transforms the operand (argument) to operator (function). It is used essentially to analyze nouns of the Korean as the operators.

Let us analyze the following sentence including the five major cases:
(1) *Gyosil-eso, Sumi-ga Minju-ege na-ege chaek-eul ju-oess-da*
Class-LOC Sumi-NOM Minju-DAT me-GEN book-ACC give–PS-DC
In the class, Sumi gave my book to Minju.

1.[N:Gyosil]-[(S/S)\N):-eso]-[N:Sumi]-[N*\N:-ga]-[N:Minju]-[N*\N:-eke]-[N:na]-[(N*/N*)\N:-uy]-[N:chaek]-[N*\N:-ul]-[X^3: ju-aes'-da]	
2. [S/S: -eso Gyosil]- [N:Sumi]-[N*\N:-ga]-[N:Minju]-[N*\N:-eke]-[N:na]-[(N*/N*)\N:-uy]- [N:chaek]-[N*\N:-ul]- [X^3: ju-aes'-da]	(<)
3. [S/S:-eso Gyosil]- [N*:-ga Sumi]- [N:Minju]-[N*\N:-eke]-[N:na]-[(N*/N*)\N:-uy]-[N:chaek]-[N*\N:-ul]- [X^3: ju-aes'-da]	(>)
4. [S/S:-eso Gyosil]-[N*:-ga Sumi]-[N*:-eke Minju]-[N:na]-[(N*/N*)\N:-uy]-[N:chaek]-[N*\N:-ul]-[X^3: ju-aes'-da]	(>)
5. [S/S:-eso Gyosil]- [N*:-ga Sumi]- [N*:-eke Minju]- [N*/N*: -uy na]- [N:chaek]-[N*\N:-ul]- [X^3: ju-aes'-da]	(>)
6. [S/S:-eso Gyosil]- [N*:-ga Sumi]- [N*:-eke Minju]- [N*/N*: -uy na]- [N*: -ul chaek]- [X^3: ju-aes'-da]	(>)
7. [S/S:-eso Gyosil]- [N*:-ga Sumi]- [N*:-eke Minju]- [N*/N*: -uy na]- [N*: -ul chaek]- [X^3: ju-aes'-da]	(>)
8. [S/S:-eso Gyosil]- [N*:-ga Sumi]- [N*: -eke Minju]- [N*: ((-uy na)-ul chaek)]- [X^3: ju-aes' - da]	(>)
9. [S/S:-eso Gyosil]- [S/X^1: C*-ga Sumi]- [N*: -eke Minju]- [N*: ((-uy na)-ul chaek)]- [X^3: ju-aes'-da]	(>T)
10. [S/S:-eso Gyosil]- [S/X^1: C*-ga Sumi]- [X^1/X^2: C*-eke Minju]- [N*: ((-uy na)-ul chaek)]- [X^3: ju-aes'-da]	(>T)
11. [S/S:-eso Gyosil]- [S/X^1: C*-ga Sumi]- [X^1/X^2: C*-eke Minju]- [X^2/X^3: C*((-uy na)-ul chaek)]- [X^3: ju-aes'-da]	(>T)
12 [S/S:-eso Gyosil]- [S/X^2: **B**((C*-ga Sumi)(C*-eke Minju))]- [X^2/X^3: C*((-uy na)-ul chaek)]- [X^3: ju-aes'-da]	(>**B**)
13. [S/S:-eso Gyosil]- [S/X^3: (**B**(**B**((C*-ga Sumi)(C*-eke Minju)))(C*((-uy na)-ul chaek))]- [X^3: ju-aes'-da]	(>**B**)
14. [S/S:-eso Gyosil]- [S: ((**B**(**B**((C*-ga Sumi)(C*-eke Minju)))(C*((-uy na)-ul chaek))ju-aes' - da)]	(>)
15. [S:(-eso Gyosil ((B(B((C*-ga Sumi)(C*-eke Minju)))(C*((-uy na)-ul chaek))ju-aes'-da))]	**(>)**
16.[S:(-eso Gyosil ((B(C*-ga Sumi)(C*-eke Minju))((C*((-uy na)-ul chaek)))ju-aes'-da))]	**(B)**
17.[S:(-eso Gyosil ((C*-ga Sumi)((C*-eke Minju)((C*((-uy na)-ul chaek))ju-aes'-da)))]	**(B)**
18.[S:(-eso Gyosil ((C*-eke Minju)(((C*((-uy na)-ul chaek))ju-aes'-da))-ga Sumi)]	**(C*)**
19.[S:(-eso Gyosil ((((C*((-uy na)-ul chaek))ju-aes'-da)-eke Minju)-ga Sumi)]	**(C*)**
20.[S:(-eso Gyosil((((ju-aes'-da)((-uy na)-ul chaek))-eke Minju)-ga Sumi))]	**(C*)**

3 Morpho-syntactical analysis in the extended Combinatory Categorial Grammar

3.1 Case system in Korean

The Korean is an agglutinative language in which the words are formed by the linking of affixes to a radical such as the cases (or postpositions). In the syntactic and semantic analysis of the Korean sentence, the cases determine the grammatical roles of nominal phrases (Sung 1999, Hong 1999, Nam 2001).

We show in the above analysis that the categorical calculus of the given sentence (1) allows us, on one hand, to verify the correct syntactic structure of the sentence by obtaining the result "S" at step 15, and on the other hand, to obtain an applicative expression that underlies this sentence structure. Furthermore, this kind of analysis allows us to deduce the syntactic types of the used cases as follows:

3.2 Morpho-Syntactical Analysis in using ACCG Parser

ACCG Parser (Applicative Combinatory Categorial Grammar Parser) is used as a formal tool to make morpho-syntactic analyses and generate underlying operator-operand or applicative representations in the model CAG. We use the applicative forms associated to the sentences, which are necessary sources for

Table 1. Syntactic types of case markers

Case		Examples	Syntactic types
Nominative		*-i/ga, -eun/neun, -kkeseo, -eso*	$(S/X^1)\backslash N$
Accusative		*-eul/reul*	$(X^1/X^2)\backslash N$, $(X^2/X^3)\backslash N$
Dative		*-ege, -kke, -hante, -bogo, -deoreo*	$(X^1/X^2)\backslash N$
Genitive		*-uy*	$(N/N)\backslash N$
Adverbials	**Place**	*-e, -eso, -eul/reul*	$(X^1/X^1)\backslash N$
	Depart	*-eso, -eul/reul*	$(X^1/X^1)\backslash N$
	Direction	*-e, -lo/eulo*	$(X^1/X^1)\backslash N$
	Goal of action	*-e, -lo/eulo, -eul/reul*	$(X^1/X^1)\backslash N$
	Quality	*-e, -lo*	$(X^1/X^1)\backslash N$
		-eulo	$(X^2/X^2)\backslash N$
	Time	*-e*	$(X^1/X^1)\backslash N$, $(S/S)\backslash N$
		-eul/reul	$(X^1/X^1)\backslash N$
	Instrument	*-lo/eulo*	$(X^1/X^1)\backslash N$
	Situation	*-e, -lo/eulo*	$(X^1/X^1)\backslash N$
	Cause	*-e, -lo/eulo*	$(X^1/X^1)\backslash N$
Vocative		*-a/ya, -yeo*	$(S/X^1)\backslash N$

more semantic and cognitive analyses. This categorial parser, ACCG[2], is based on the Applicative Combinatory Categorial Grammar [1], which is an extended version of the Categorial Grammars [7,9]. It uses combinators of the CL.

We wrote the algorithms of the categorial calculi (on syntactic functional types) with an implementation of applicative combinatory categorial rules and meta-rules (Biskri, 1995; Kang, Desclés, 2008). We present the results of the sentence (2) obtained by the ACCG Parser (Figure 2).

(2)장이 거기에서 책을 읽고 있었죠.
Jean-i geogi-eso chaek-eul il-go iss-eoss-jo
John-Nom there-Place book-Acc read-Comp. is-Ps-Nar.
John was reading a book there

The applicative representations marked in Figure 2 show the structures of operator/operand of the example sentences. This result is the applicative tree generated from the applicative representation. Comparing the results obtained by other categorical parsers, such as CCG Parser [3] and POSPAR [2], ACCG Parser offer linguistically correct analyses (See [6] for the comparison of the systems).

[2] The ACCG Parser runs to give a syntactic analysis in French, in English and in Korean, but in this paper, we focus on the results obtained in Korean.

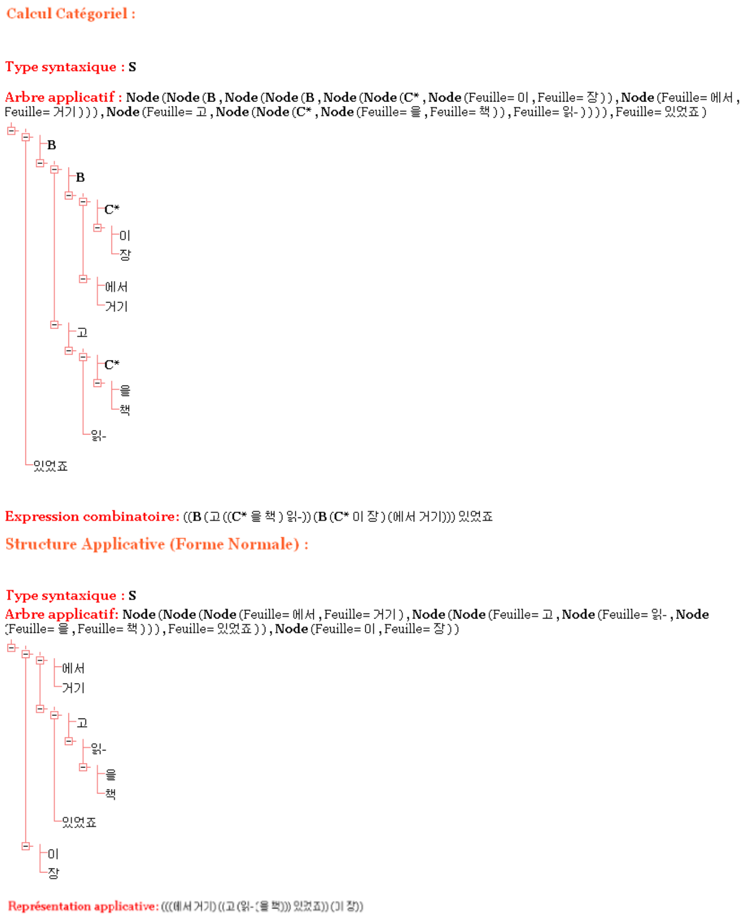

Fig. 2. Result obtained by ACCG Parser

4 Conclusion and Future Work

As we have shown in this paper, an extended Categorial Grammar such as ACCG allows us to scope the difficult characteristics of the Korean language. In particular, we could consider the cases in Korean as operators which play an essential role in the Korean analysis and give the automatic morhpo-syntactical analysis with the ACCG parser. But the data was not enough to make a serious evaluation of this parser. So we plan to evaluate the system using more numerous texts and to find some possibilities of its application to other languages such as Czech.

Acknowledgements This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536 and by the Czech Science Foundation under the project P401/10/0792.

References

1. Biskri, I. and Desclés, J.-P. (2005). Applicative and Combinatory Categorial Grammar and Subordinate Constructions in French. *International Journal on Artificial Intelligence Tools*, Vol 14, No. 1& 2: 125-136.
2. Cha, J.W. , Lee, G. and Lee, J.H. (2002). Korean Combinatory Categorial Grammar and statistical parsing. Computers and the Humanities, 36(4):431-453.
3. Clark, S. and Curran, J. R. (2004). Parsing the WSJ using CCG and log-linear models. In *Proceedings of the 42nd Meeting of the ACL*, 104-111.
4. Curry, H.B. and Feys, R. (1958). *Combinatory Logic*. Volume 1. Amsterdam: North-Holland Publishing.
5. Desclés, J.-P. (2004). Combinatory Logic, Language and Cognitive Representations. In Weingartner P. (eds), *International Meeting on Alternative Logics*, Springer: 115-148.
6. Kang, J., Desclés, J.-P. (2011). ACCG : un système catégoriel d'analyse automatique de constructions syntaxiques. *ACFAS*. 9-13 mai.
7. Oehrle, R.T., Bach, E., Wheeler, D. (1988). *Categorial Grammars and Natural Languages Structures*. D. Reidel, Boston, MA.
8. Shaumyan, S. K. (1987). *A Semiotic Theory of Natural Languages*, Bloomington, Indiana Univ. Press.
9. Steedman, M. (2000). *The Syntactic Process*. MIT Press/Bradford Books, Cambridge, MA.