ABSTRACT OF THE THESIS

T i t l e : A Computational Treatment of V-V Compounds in Japanese
Adviser : GUNJI Takao
Author : HASHIMOTO Chikara
D a t e : November, 2004

The purposes of the thesis are to implement the linguistic analyses of Japanese verbal compounds in a computational grammar of Japanese and to discuss why and how Natural Language Processing (NLP) should benefit from theoretical linguistics.

In chapter 1, I describe the difference between theoretical linguistics and NLP, and then I argue that NLP should make use of linguistics on the basis that we can acquire a fine-grained semantic representation by means of a deep linguistic treatment and that a linguistic treatment of NLP do not have to rely as heavily on statistical information, as long as a grammar describes a language precisely. Japanese verbal compounds (V_1-V_2 compounds) are one kind of **Multiword Expressions** (**MWEs**) (Sag et al., 2002), which NLP researchers have recently acknowledged as an annoying problem. As such, V_1-V_2 compounds resist simple solutions. If we regard all MWEs as totally compositional, and derive all of them by means of some sort of rule, we would face **overgeneration problem** and **idiomaticity problem**; that is, we would overgenerate unattested V_1-V_2s and cannot treat V_1-V_2's idiomaticity. On the other hand, if we regard all MWEs as single words, and register all of them in the lexicon, then we would suffer from **flexibility problem** and **lexical proliferation problem**; namely we would suffer from V_1-V_2's productivity. These constitute the evidences that we definitely need a sophisticated linguistic analysis to deal with V_1-V_2 compounds.

In chapter 2, I first describe the criteria of Hasida (1997) by which a linguistic theory is judged to be suitable for NLP. The criteria include **Importance of Phenomena**, whether the problem that a linguistic theory tries to account for is also important for NLP, **Simplicity of Design**, whether a theory makes a NLP system simple, **Efficiency of Computation**, whether computation posited by a theory is executed by computer efficiently, **Availability of Input**, whether inputs that a theory makes reference to are easily available for NLP systems. Next I move on to the critique of Kageyama (1993) and Matsumoto (1996) in light of Hasida (1997), although my analyses owe much to them.

Based on the GB theory, Kageyama (1993) distinguishes **syntactic V_1-V_2 compounds** and **lexical V_1-V_2 compounds**. He further divides syntactic V_1-V_2s into Raising, Control, and $\overline{V}$ complementation types. Regarding lexical V_1-V_2s, he proposes the Transitivity Harmony Principle, and posits the back formation analysis and the LCS analysis for some exceptions to the principle. Although Kageyama's (1993) analysis gives us a theoretical basis of computational implementation of V_1-V_2 compounds, it has several defects in terms of Hasida (1997); the GB analyses, especially the movement analysis and the empty category analysis, lack a mathematical foundation, and thus lacks efficient processing techniques, resulting in the violation of Efficiency of Computation. In addition, his analysis of lexical V_1-V_2s violates Simplicity of Design and Importance of Phenomena, since he posits computationally expensive machinery to account for a few exceptions.

Matsumoto (1996) presents comprehensive and suggestive observations about lexical V_1-V_2s based on argument structure. He classifies lexical V_1-V_2s into Pair compounds, Cause compounds, Manner compounds, Means compounds, Compounds exhibiting other relations, Compounds with semantically deverbalized V_2, and Compounds with semantically deverbalized V_1, and tries to analyze them in terms of a semantic relation between V_1 and V_2. However, recognizing such a semantic relation involves pragmatics or world knowledge, which means that it would be difficult for computers to do such a job. In other words, the analysis of Matsumoto (1996) violates Availability of Input in that it refers to information that a computer cannot easily obtain. As well, semantic notions that his lexical analysis makes use of are too fine-grained for us to develop a large-scale grammar and lexicon, resulting in the violation of Simplicity of Design.

Through this chapter, it is shown that a sophisticated linguistic analysis is indispensable for a computational treatment of V_1-V_2 compounds, since they show complicated MWEs characteristics.

In chapter 3, I present my analyses of V_1-V_2 compounds. I first describe my policy of grammar development that observes the criteria of Hasida (1997). In order to satisfy Importance of Phenomena, I avoid complicating my analyses to account for exceptional cases and linguistic phenomena that people's judgments are not stable or consistent. Also, to satisfy Simplicity of Design, I make my analyses descriptively adequate rather than theoretically advanced, even though they would not be parsimonious. Availability of Input is met by restricting information that is referred to by my analyses to those that are computationally available. As for Efficiency of Computation, I adopt the $\mathcal{TDL}$ language (Krieger & Schafer, 1994) as a grammar description language so that my analyses would be executed efficiently.

I also describe the framework of my analyses. I implement my analyses on the existing computational grammar of Japanese, JACY (Siegel & Bender, 2002), which adopts **Head-driven Phrase Structure Grammar** (**HPSG**, Sag and Wasow (1999)) as a syntactic framework and **Minimal Recursion Semantics** (**MRS**, Copestake et al. (1999)) as a semantic framework. In the implementation, I use the **LKB** system (Copestake, 2002).

My analysis of syntactic V_1-V_2s roughly follows Kageyama (1993), and I classify syntactic V_1-V_2s into **A type**, **B type**, and **C type** (Hashimoto, 2003). In particular, I posit VP embedding structures for A and B type. The structure is indispensable for the theoretically precise analyses for them, although almost all of the previous computational grammars of Japanese have avoided it because of a difficulty involving scrambling. As a result, I can acquire a fine-grained semantic representation, which is essential to a precise NLP. Besides, my analysis is a simple phrase structure analysis without movement nor empty categories, and still it is theoretically precise. That way, my analysis satisfies Efficiency of Computation. However, the VP embedding structures cause a problem concerning scrambling. To get around the problem, I posit **Argument Attraction**, which is precise and properly restricted. I discuss the approach is more efficient than alternative approaches thanks to its restrictive nature.

Roughly following Matsumoto (1996), I classify lexical V_1-V_2s into **Right headed V_1-V_2**, **Argument mixing V_1-V_2**, **V_1-V_2 with semantically deverbalized V_1**, **V_1-V_2 with semantically deverbalized V_2**, and **Non-compositional V_1-V_2**. Right headed V_1-V_2 and Argument mixing V_1-V_2 cover Pair, Cause, Manner, and Means compounds of Matsumoto (1996), but I underspecify the four semantic relations. This strategy is

justifiable on the ground of Availability of Input. My analysis of lexical V_1-V_2s is simple and is based on argument structure of Imaizumi and Gunji (2000). Previous computational grammars of Japanese have avoided adopting argument structure, but it is also essential to theoretical preciseness. Thanks to the conciseness and the argument structure, my analysis better satisfies Simplicity of Design. In addition, it successfully accounts for lexical V_1-V_2's syntactic and semantic properties. Especially, we can acquire the correct semantic representation of lexical V_1-V_2s, as well as that of syntactic V_1-V_2s.

Through chapter 3, it is shown that my analyses capture the MWEs properties of V_1-V_2 compounds while observing the criteria of Hasida (1997). Notably, the VP embedding structures and argument structure play a important role.

In chapter 4, I describe the evaluation experiment through which I illustrate the coverage, the number of ambiguity and the efficiency of my implementation. In the evaluation, I used the [incr tsdb()] system (Oepen & Carroll, 2000) and the Lexeed corpus (Kasahara et al., 2004). I also prepared two versions of JACY: JACY-vv and JACY-plain. JACY-vv is equipped with my implementation, but is not given lexical entries for V_1-V_2s except for those of non-compositional V_1-V_2s. On the other hand, JACY-plain, which is the original one, has no rule for V_1-V_2s, but contains 1,325 lexical entries for V_1-V_2s in the lexicon. Consequently, JACY-vv outperformed JACY-plain in terms of coverage and the number of ambiguity. The more coverage was gained because of the remarkably high productivity of V_1-V_2s. JACY-vv, but not JACY-plain, could deal with it. In other words, JACY-vv could get around the lexical proliferation problem; it can handle the unknown V_1-V_2s by means of appropriate rules. On the other hand, the 1,325 entries of JACY-plain, which was not quite small, could not deal with the productivity. The reason for the less ambiguity involves the difference of the treatment of scrambling from an embedded VP. To be more precise, the restrictive nature of my Argument Attraction approach made us get less ambiguity. Also, since JACY-vv distinguishes productive V_1-V_2s from non-productive ones and compositional V_1-V_2s from non-compositional ones, it can get around the overgeneration problem and the idiomaticity problem. However, as for performance, JACY-vv turned out to be working less efficiently than JACY-plain. Generally, more rules lead to less efficiency, but I discuss the possibility that changing grammatical representations would make the grammar more efficient.

In chapter 5, I first summarize the contents from chapter 1 to chapter 4, then I discuss future works and the prospect of the relationship between theoretical linguistics and NLP. The future works include how we treat V_1-V_2s that the current analyses cannot deal with, how we automatically detect non-compositional V_1-V_2s, and how we make computers translate Japanese V_1-V_2s into English expressions. Regarding the treatment of problematic V_1-V_2s, I claim that, first of all, we should find how productive they are through a corpus study. If they are really productive, we should add new rules to deal with them. Otherwise, we should enter them in the lexicon as single words. As for the automatic detection of non-compositional V_1-V_2s, I take up the studies on the automatic detection of English phrasal verbs, and discuss the applicability of the studies to Japanese V_1-V_2 compounds. Finally, I discuss the prospect of the two studies of language: theoretical linguistics and NLP. I mention several NLP problems that theoretical linguistics cannot help. Then I discuss how NLP contributes to the resolution of the biggest issues of linguistics, and advocate a deep linguistic NLP.

References

Copestake, A. (2002). *Implementing Typed Feature Structure Grammars*. CSLI Publications.

Copestake, A., Flickinger, D. P., & Sag, I. A. (1999). Minimal Recursion Semantics: An Introduction. Manuscript, Stanford University: CSLI.

Hashimoto, C. (2003). HPSG analysis of Long Distance Passives (in Japanese). In *Proceedings of the 126th annual meeting of Linguistic Society of Japan*, pp. 256–261 Tokyo, Japan.

Hasida, K. (1997). Information Science Approach to Language. In Ôtsu, Y., Gunji, T., Takubo, Y., Nagao, M., Hasida, K., Masuoka, T., & Matsumoto, Y. (Eds.), *An Introduction to Language Science* (in Japanese), chap. 3. Iwanami.

Imaizumi, S., & Gunji, T. (2000). Complex Events in Lexical Compounds. In Itou, T., & Yatabe, S. (Eds.), *Lexicon and Syntax* (in Japanese), pp. 33–59. Hitsuji Shobou.

Kageyama, T. (1993). *Grammar and Word Formation* (in Japanese). Hitsuji Shobou.

Kasahara, K., Sato, H., Bond, F., Tanaka, T., Fujita, S., Kanasugi, Y., & Amano, S. (2004). Construction of a Japanese Semantic Lexicon: Lexeed. In *Information Processing Society of Japan, 2004-NL-159*, pp. 75–82 Tokyo, Japan.

Krieger, H.-U., & Schafer, U. (1994). $\mathcal{TDL}$ — A type description language for constraint-based grammars. In *Proceedings of the 15th International Conference on Computational Linguistics*.

Matsumoto, Y. (1996). *Complex Predicates in Japanese: A Syntactic and Semantic Study of the Notion 'Word'*. CSLI Publications.

Oepen, S., & Carroll, J. (2000). Performance profiling for grammar engineering. *Natural Language Engineering*, 81–97.

Sag, I. A., Baldwin, T., Bond, F., Copestake, A., & Flickinger, D. (2002). Multiword expressions: A pain in the neck for NLP. In *Computational Linguistics and Intelligent Text Processing: Third International Conference*, pp. 1–15 Mexico City, Mexico.

Sag, I. A., & Wasow, T. (1999). *Syntactic Theory: A Formal Introduction*. Center for the Study of Language and Information, Stanford. Japanese edition (two volumes) – translated and edited by Takao Gunji and Yasunari Harada, appeared in 2001.

Siegel, M., & Bender, E. M. (2002). Efficient Deep Processing of Japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization* Taipei, Taiwan.