Compilation of a Dictionary of Japanese Functional Expressions with Hierarchical Organization

Suguru Matsuyoshi[1], Satoshi Sato[1], and Takehito Utsuro[2]

[1] Graduate School of Engineering, Nagoya University,
Chikusa, Nagoya, 464-8603, Japan
[2] Graduate School of Systems and Information Engineering, University of Tsukuba,
Tennodai, Tsukuba, 305-8573, Japan

Abstract. The Japanese language has a lot of functional expressions, which consist of more than one word and behave like a single functional word. A remarkable characteristic of Japanese functional expressions is that each functional expression has many different surface forms. This paper proposes a methodology for compilation of a dictionary of Japanese functional expressions with hierarchical organization. We use a hierarchy with nine abstraction levels: the root node is a dummy node that governs all entries; a node in the first level is a headword in the dictionary; a leaf node corresponds to a surface form of a functional expression. Two or more lists of functional expressions can be integrated into this hierarchy. This hierarchy also provides a way of systematic generation of all different surface forms. We have compiled the dictionary with 292 headwords and 13,958 surface forms, which covers almost all of major functional expressions.

1 Introduction

Some languages have *functional expressions*, which consist of more than one word and behave like a single functional word. In English, "in spite of" is a typical example, which behaves like a single preposition. In natural language processing (NLP), correct detection of functional expressions is crucial because they determine sentence structures and meanings. Implementation of a detector of functional expressions requires a dictionary of functional expressions, which provides lexical knowledge of every functional expression.

The Japanese language has many functional expressions. They are classified into three types according to the classification of functional words: particle, auxiliary verb, and conjunction. The particle type is sub-divided into five sub-types: case-marking particle, conjunctive particle, adnominal particle, focus particle, and topic-marking particle. A remarkable characteristic of Japanese functional expressions is that each functional expression has many different surface forms; they include *derivations*, *expression variants* produced by particle alternation and insertion, *conjugation forms* produced by the final conjugation component, and *spelling variants*.

Y. Matsumoto et al. (Eds.): ICCPOL 2006, LNAI 4285, pp. 395–402, 2006.

Compilation of a dictionary of Japanese functional expressions for natural language processing requires two lists. The first is a list of headwords of the dictionary; the second is the complete list of surface forms of entries in the first list.

Although there are several lists of Japanese functional expressions such as [2] and [3], compilation of the first list is not straightforward because there is no concrete agreement on the selection guideline of headwords of Japanese functional expressions. For example, [2] and [3] follow different selection guidelines: both of "にたいして (ni-taishi-te)" and "にたいする (ni-taisuru)" are headwords in [2]; only the former is a headword and the latter is its derivation in [3]. We need a way of resolving this type of contradiction to merge different lists of headwords.

The second list is required because NLP systems have to process functional expressions in surface forms that appear in actual texts. Because native speakers easily identify functional expressions in surface forms, there is no explicit list that enumerates all surface forms in dictionaries for human use. We need a systematic way of generating the complete list of surface forms for machine use.

This paper proposes a methodology for compilation of a dictionary of Japanese functional expressions with hierarchical organization. We design a hierarchy with nine abstraction levels. By using this hierarchy, we can merge different lists of headwords, which are compiled according to different guidelines. This hierarchy also provides a way of systematic generation of all different surface forms.

2 Hierarchical Organization of Functional Expressions

2.1 Various Surface Forms of Japanese Functional Expressions

Several different language phenomena are related to the production of various surface forms of Japanese functional expressions. We classify these surface-form variants into four categories: *derivations*, *expression variants*, *conjugation forms*, and *spelling variants*.

In case two forms that have different grammatical functions are closely related to each other, we classify them into *derivations*. For example, "にたいする (ni-taisuru)" and "にたいして (ni-taishi-te)" are closely related to each other because "たいする (taisuru)" and "たいして (taishi-te)" are different conjugation forms of the same verb. They have different grammatical functions: the former behaves like an adnominal particle and the latter behaves like a case-marking particle. Therefore we classify them into derivations. This view comes from the fact that several case-marking particles can be used as adnominal particles with slightly different forms.

In case two forms have slightly different morpheme sequences with the same grammatical function and meaning except style (formal or informal), we classify them into *expression variants*. Language phenomena that are related to production of expression variants are:

1. Alternation of functional words (particles and auxiliary verbs)
   In a functional expression, a component functional word may be replaced by another functional word with the same meaning. For example, "からすれば (kara-sure-ba)" is produced from "からすると (kara-suru-to)" by substitution of "ば (ba)" for "と (to)," where these two particles have the same meaning (assumption).
2. Phonetic phenomena
   (a) Phonetic contraction
   For example, "なけりゃならない (nakerya-nara-nai)" is produced from "なければならない (nakere-ba-nara-nai)," where "りゃ (rya)" is a shorter form of "れば (re-ba)," which is produced by phonetic contraction.
   (b) Ellipsis
   In case a component word has an informal (ellipsis) form, it may be replaced by the informal form. For example, "とこだった (toko-daQ-ta)" is produced from "ところだった (tokoro-daQ-ta)," where "とこ (toko)" is an informal form of "ところ (tokoro)," which is produced by omission of "ろ (ro)."
   (c) Voicing
   The initial consonant "*t*" of a functional expression may change to "*d*," depending on the previous word. For example, "ていい (te-ii)" changes into "でいい (de-ii)" when it occurs just after "読ん (yoN)."
3. Insertion of a focus particle
   A focus particle such as "は (ha)" and "も (mo)" [4] can be inserted just after a case-marking particle. For example, "とはいっても (to-ha-iQ-te-mo)" is produced from "といっても (to-iQ-te-mo)" by insertion of "は (ha)" just after "と (to)."

The third category of surface-form variants is *conjugation forms*. In case the last component of a functional expression is a conjugation word, the functional expression may have conjugation forms in addition to the base form. For example, a functional expression "ことにする (koto-ni-suru)" has conjugation forms such as "ことにし (koto-ni-shi)" and "ことにすれ (koto-ni-sure)," because the last component "する (suru)" is a conjugation word.

Some conjugation forms have two different forms: the normal conjugation form and the *desu/masu* (polite) conjugation form. For example, a variant "ことにします (koto-ni-shi-masu)" is the *desu/masu* conjugation form of "ことにする (koto-ni-suru)," where "します (shi-masu)" is the *desu/masu* form of "する (suru)."

The last category of surface-form variants is *spelling variants*. In Japanese, most words have *kanji* spelling in addition to *hiragana* spelling. For example, both of "にあたって (ni-ataQ-te)" (hiragana spelling) and "に当たって (ni-ataQ-te)" (kanji spelling) are used in practice.

2.2 Hierarchy with Nine Abstraction Levels

In order to organize functional expressions with various surface forms described in the previous subsection, we design a hierarchy with nine abstraction levels. Figure 1 shows a part of the hierarchy. In this hierarchy, the root node (in L^0) is a dummy node that governs all entries in the dictionary. A node in L^1 is an entry (headword) in the dictionary; the most generalized form of a functional expression. A leaf node (in L^9) corresponds to a surface form (completely-instantiated form) of a functional expression. An intermediate node corresponds to a partially-abstracted (partially-instantiated) form of a functional expression.

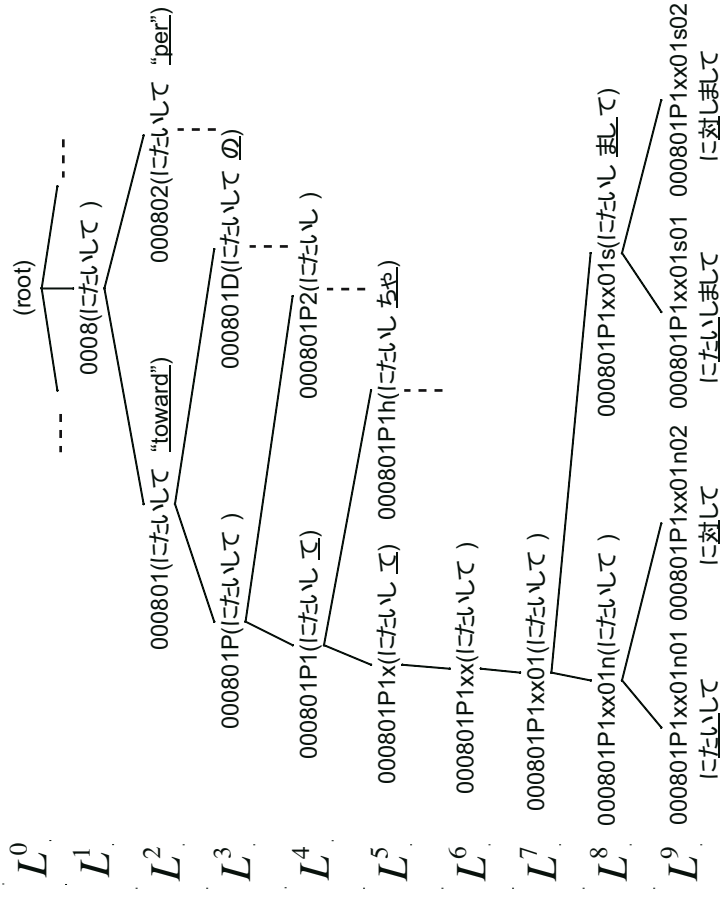

Fig. 1. A part of the hierarchy

Table 1 overviews the nine abstraction levels of the hierarchy. From L^3 to L^9 correspond to the phenomena described in the previous subsection. First, we have defined the following order according to the significance of categories of surface-form variants:

derivations (L^3) > expression variants (L^4, L^5, L^6)
> conjugation forms (L^7, L^8) > spelling variants (L^9) .

Then, we have defined the order L^4–L^6 and L^7–L^8 in order to make a simple hierarchy.

Table 1. Nine abstraction levels

Abstraction Levels		ID		Number of Nodes
		Character Type	Length	
L^1	Headword	digit	4	292
L^2	Meaning categories	digit	2	354
L^3	Grammatical functions	8 alphabets	1	470
L^4	Alternations of functional words	digit	1	682
L^5	Phonetic variations	32 alphabets	1	1,032
L^6	Optional focus particles	17 alphabets	1	1,628
L^7	Conjugation forms	digit	2	6,190
L^8	Normal or *desu*/*masu* forms	2 alphabets	1	8,462
L^9	Spelling variations	digit	2	13,958

In addition to these seven levels, we define the following levels.

L^2 Meaning categories
Several functional expressions take more than one meaning. For example, "にたいして (ni-taishi-te)" takes two different meanings. The first meaning is "toward"; e.g., "彼は私にたいして親切だ" (He is kind toward me). The second meaning is "per"; e.g., "一人にたいして５つ" (five per one person). This level is introduced to distinguish such ambiguities.

L^1 Headword
A node of this level corresponds to a headword of the dictionary.

Because the hierarchy covers from the most generalized form (in L^1) of a functional expression to the completely-instantiated forms (in L^9) of it, any form of a functional expression can be inserted in some position in the hierarchy.

From this hierarchy, multiple lists of headwords can be generated. Our list of headwords is nodes in L^1. In case you follow the guideline that each headword has the unique meaning, which roughly corresponds to the guideline used by the book [3], nodes in L^2 become headwords. In case you follow the guideline that each headword has the unique grammatical function, nodes in L^3 become headwords.

We design an ID system in which the structure of hierarchy can be encoded; an ID consists of nine parts, each of which corresponds to one of nine levels of the hierarchy (in Fig. 2). We assign a unique ID to each surface form. Because an ID represents the position of the hierarchy, we easily obtain the relation between two surface forms by comparing their IDs. Table 2 shows three surface forms

of functional expressions. By comparing IDs of (1) and (2), we obtain that the leftmost difference is “x” and “h” at the ninth character; it corresponds to L^5 so they are phonetic variants of the same functional expression. In contrast, the first 4 digits are different between (1) and (3); from this, we obtain that they are completely different functional expressions.

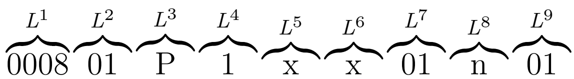

Fig. 2. An ID consists of nine parts

Table 2. Functional expressions with similar IDs

	ID	Functional Expression
(1)	000801P1xx01n01	にたいして (ni-taishi-te)
(2)	000801P1hx01n01	にたいしちゃ (ni-taishi-cha)
(3)	000901P1xx01n01	について (ni-tsui-te)

3 Compilation of a Dictionary of Functional Expressions

3.1 Compilation Procedure

We have compiled a dictionary of Japanese functional expressions, which has the hierarchy described in the previous section. The compilation process is incremental generation of the hierarchy, because we have neither the complete list of headwords nor the list of all possible surface forms in advance.

The compilation procedure of an incremental step is:

1. Pick up a functional expression from [3].
2. Create a node that corresponds to the given expression and insert it at the appropriate position of the hierarchy.
3. Create the lower subtree under the node.

Most of headwords in [3] correspond to nodes in L^2. Some exceptions correspond to nodes in L^4 or L^5. In order to insert such nodes into the hierarchy, we create the additional upper nodes if necessary.

In step 3, we create the lower subtree under the inserted node, which means enumeration of all possible surface forms of the functional expression. Most of surface forms can be generated automatically by applying generation templates to the inserted node. We manually remove overgenerated (incorrect) forms from the generated subtree. Several exceptional forms are not included in the generated subtree. We manually add such exceptional forms into the subtree.

We have already inserted 412 functional expressions, which are all functional expressions described in [3], into the hierarchy. The number of nodes in each level is shown in Table 1. The number of nodes in L^1 (headwords) is 292, and the number of leaf nodes (surface forms) is 13,958.

3.2 Description of Functional Expressions

When we create a leaf node in the hierarchy, we assign the following eight properties to the node.

1. ID (described in Sect. 2.2)
2. Meaning category
   We employ 103 meaning categories to describe meanings of functional expressions and to distinguish ambiguities in meaning (in L^2). We specify one of them in this slot.
3. Readability
   Some functional expressions are basic, i.e., everyone knows them; some are not. In this slot, we specify one of readability levels of A1, A2, B, C, and F, where A1 is the most basic level and F is the most advanced level.
4. Style
   We specify one of four styles: normal, polite, colloquial, and stiff.
5. Negative expressions
   We specify expressions that have an opposite meaning against the functional expression. These are required because literally negative forms of functional expressions may be ungrammatical.
6. Idiomatic expressions that include the functional expression
7. Example sentences
8. Reference

In practice, we specify the above properties at the appropriate intermediate nodes in the hierarchy, not at leaf nodes. For example, we specify meaning categories at nodes in L^2; we specify styles at nodes in L^8. A standard inheritance mechanism automatically fills all slots in the leaf nodes. This way of specification clarifies the relation between properties and forms of functional expressions; e.g., the style property is independent of spelling variants.

4 Related Work

There is no large electronic dictionary of Japanese functional expressions that is available in public.

Shudo et al. have collected 2,500 functional expressions in Japanese (1,000 of particle type and 1,500 of auxiliary-verb type) and classified them according to meaning [5,6]. In the list, the selection of headwords is not consistent, i.e., headwords of different abstraction levels exist; they correspond to the nodes at L^3, L^4, and L^5 in our dictionary. This list has no explicit organization structure except alphabetic order.

Hyodo et al. have proposed a dictionary of Japanese functional expressions with two layers [1]. This dictionary has 375 entries in the first layer: from these entries, 13,882 surface forms (in the second layer) are generated automatically. This dictionary does not provide precise classification between two surface forms, such as phonetic variants and spelling variants, which our dictionary provides.

5 Conclusion and Future Work

We have proposed a methodology for compilation of a dictionary of Japanese functional expressions with hierarchical organization. By using this methodology, we have compiled the dictionary with 292 headwords and 13,958 surface forms. It covers all functional expressions described in [3]. The compilation process of integrating additional functional expressions, which are described in [2], not in [3], is planned in the next step.

Our dictionary can be used for various NLP tasks including parsing, generation, and paraphrasing of Japanese sentences. For example, the use of our dictionary will improve the coverage of the detection method of functional expressions [7]. Experimental evaluation of application of this dictionary to actual NLP tasks is future work.

References

1. Yasuaki Hyodo, Yutaka Murakami, and Takashi Ikeda. A dictionary of long-unit functional words for analyzing *bunsetsu*. In *Proceedings of the 6th Annual Meeting of the Association for Natural Language Processing*, pages 407–410, 2000. (in Japanese).
2. Group Jamashii, editor. Nihongo Bunkei Jiten *(Dictionary of Sentence Patterns in Japanese)*. Kurosio Publisher, 1998. (in Japanese).
3. Yoshiyuki Morita and Masae Matsuki. Nihongo Hyougen Bunkei, *volume 5 of* NAFL Sensho *(Expression Patterns in Japanese)*. ALC Press Inc., 1989. (in Japanese).
4. Yoshiko Numata. Toritateshi (*Focus Particles*). In Keiichiro Okutsu, Yoshiko Numata, and Takeshi Sugimoto, editors, Iwayuru Nihongo Joshi no Kenkyu *(Studies on So-called Particles in Japanese)*, chapter 2. BONJINSHA, 1986. (In Japanese).
5. Kosho Shudo, Toshiko Narahara, and Sho Yoshida. Morphological aspect of Japanese language processing. In *Proceedings of the 8th COLING*, pages 1–8, 1980.
6. Kosho Shudo, Toshifumi Tanabe, Masahito Takahashi, and Kenji Yoshimura. MWEs as non-propositional content indicators. In *Proceedings of the 2nd ACL Workshop on Multiword Expressions: Integrating Processing (MWE-2004)*, pages 32–39, 2004.
7. Masatoshi Tsuchiya, Takao Shime, Toshihiro Takagi, Takehito Utsuro, Kiyotaka Uchimoto, Suguru Matsuyoshi, Satoshi Sato, and Seiichi Nakagawa. Chunking Japanese compound functional expressions by machine learning. In *Proceedings of the EACL 2006 Workshop on Multi-Word-Expressions in a Multilingual Context*, pages 25–32, 2006.