# Dictionary Look-up with Katakana Variant Recognition

**Satoshi Sato**

Graduate School of Engineering
Nagoya University
Chikusa-ku, Nagoya,
464-8603, JAPAN
ssato@nuee.nagoya-u.ac.jp

**Abstract**

The Japanese language has rich variety and quantity of word variant. Since 1980s, it has been recognized that this richness becomes an obstacle against natural language processing. A complete solution, however, has not been presented yet. This paper proposes a method to recognize Katakana variants—a major type of word variant in Japanese—in the process of dictionary look-up. For a given set of variant generation rules, the method executes variant generation and entry retrieval simultaneously and efficiently. We have developed the seven-layered rule set (216 rules in total) according to the specification manual of UniDic-2.1.0 and other sources. An experiment shows that the spelling-variant generator with 102 rules in the first five layers is almost perfect. Another experiment shows that the form-variant generator with all 216 rules is powerful and 77.7% of multiple spellings of Katakana loanwords are unnecessary (i.e., can be removed). This result means that the proposed method can drastically reduce the number of variants that we have to register into a dictionary in advance.

**Keywords:** word variant, Katakana variant, dictionary look-up

## 1. Introduction

There are some words that have multiple spellings. For example, in English, the word “color” (in American English) can also be spelled as “colour” (in British English). Usually, this type of variant is called *spelling variant*.

The Japanese language has rich variety and quantity of spelling variant and other types of word variant (The National Language Research Institute, 1983). This is partially caused by the Japanese writing system, which has three main scripts (character types): Hiragana, Katakana, and Kanji. Another reason is a flood of words imported from English and other languages.

Transliteration is a standard method of importing words from foreign languages into Japanese. The imported words are called *Katakana loanwords*, because they are written in Katakana characters. The portion of Katakana loanwords is growing year by year. For example, UniDic-2.1.0, a recently-compiled Japanese dictionary for morphological analysis, has 26,228 Katakana loanwords, which are 12.7% of the dictionary.

A noticeable characteristic of Katakana loanwords is richness of word variant, which is caused by nondeterminism of transliteration. There are some sounds in foreign languages that do not directly correspond to Japanese sounds and *sound approximation*, which maps such sounds to similar Japanese sounds, is necessary in transliteration. For example, 16 different variants of transliteration of “initiative” are registered in UniDic-2.1.0.

A simple and traditional method of handling word variant is registration of all variants in a dictionary. It is not a complete solution because compilation of the exhaustive list of word variants is impossible because of variant richness. A smart method, which can handle unregistered variants, is required.

This paper proposes a method to recognize Katakana variants in the process of dictionary look-up. Because this method can retrieve the correct entry from an unregistered variant, it can drastically reduce the number of variants that we have to register into a dictionary in advance.

The remaining of the paper is organized as follows. Section 2 explains why Katakana variants are produced. Section 3 describes the related work. Section 4 describes the UniDic dictionary and its three-layered structure, which provides a clear distinction of two types of word variant: spelling variant and form variant. Section 5 proposes a method of dictionary look-up with variant recognition and Section 6 describes an experimental result.

## 2. Katakana Variant

Katakana is one of three main scripts used in the Japanese writing system. Katakana is a set of phonetic characters and used for foreign words and names, loanwords, and scientific names.

### 2.1. Official Guideline

There is a short official guideline (Cabinet notifications and directives, 1991) that suggests how to spell Katakana loanwords. It consists of five rules that refer two tables, with commentary notes. The first table (shown in Table 1) defines 69 Katakana characters, 46 Katakana sequences, and 1 symbol, which are widely used to spell Katakana loanwords. The second table (shown in Table 2) defines additional 1 Katakana character and 19 Katakana sequences, which may be used to spell Katakana loanwords. Note that this guideline is not rigid; it allows the use of Katakana sequences that are not defined in these two tables as exceptions.

The first table is divided into two sub-tables: the upper table (Table-1A) and the lower table (Table-1B). Each element in Table-1A corresponds to a Japanese sound. In contrast, each element in Table-1B corresponds a non-Japanese sound, i.e., a sound imported from a foreign language.

| | | | | |
|---|---|---|---|---|
| ア (a) | イ (i) | ウ (u) | エ (e) | オ (o) |
| カ (ka) | キ (ki) | ク (ku) | ケ (ke) | コ (ko) |
| サ (sa) | シ (shi) | ス (su) | セ (se) | ソ (so) |
| タ (ta) | チ (chi) | ツ (tsu) | テ (te) | ト (to) |
| ナ (na) | ニ (ni) | ヌ (nu) | ネ (ne) | ノ (no) |
| ハ (ha) | ヒ (hi) | フ (fu) | ヘ (he) | ホ (ho) |
| マ (ma) | ミ (mi) | ム (mu) | メ (me) | モ (mo) |
| ヤ (ya) | | ユ (yu) | | ヨ (yo) |
| ラ (ra) | リ (ri) | ル (ru) | レ (re) | ロ (ro) |
| ワ (wa) | | | | |
| ガ (ga) | ギ (gi) | グ (gu) | ゲ (ge) | ゴ (go) |
| ザ (za) | ジ (ji) | ヅ (zu) | ゼ (ze) | ゾ (zo) |
| ダ (da) | | | デ (de) | ド (do) |
| バ (ba) | ビ (bi) | ブ (bu) | ベ (be) | ボ (bo) |
| パ (pa) | ピ (pi) | プ (pu) | ペ (pe) | ポ (po) |
| キャ(kya) | | キュ(kyu) | | キョ(kyo) |
| シャ(sha) | | シュ(shu) | | ショ(sho) |
| チャ(cha) | | チュ(chu) | | チョ(cho) |
| ニャ(nya) | | ニュ(nyu) | | ニョ(nyo) |
| ヒャ(hya) | | ヒュ(hyu) | | ヒョ(hyo) |
| ミャ(mya) | | ミュ(myu) | | ミョ(myo) |
| リャ(rya) | | リュ(ryu) | | リョ(ryo) |
| ギャ(gya) | | ギュ(gyu) | | ギョ(gyo) |
| ジャ(ja) | | ジュ(ju) | | ジョ(jo) |
| ビャ(bya) | | ビュ(byu) | | ビョ(byo) |
| ピャ(pya) | | ピュ(pyu) | | ピョ(pyo) |
| ン (n) | | | | |
| ッ (Sokuon) | | | | |
| ー (long vowel symbol) | | | | **(1A)** |
| | | | シェ(*she*) | |
| | | | チェ(*che*) | |
| ツァ(*tsa*) | | | ツェ(*tse*) | ツォ(*tso*) |
| | ティ(*ti*) | | | |
| ファ(*fa*) | フィ(*fi*) | | フェ(*fe*) | フォ(*fo*) |
| | | | ジェ(*je*) | |
| | ディ(*di*) | | | |
| | | デュ(*du*) | | **(1B)** |

Note: Romanized spellings in italic face are unofficial.

Table 1: The first table in the official guideline

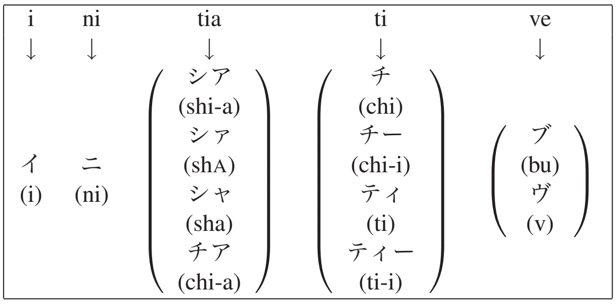
| | | | | |
|---|---|---|---|---|
| | | | イェ(*ye*) | |
| | ウィ(*wi*) | | ウェ(*we*) | ウォ(*wo*) |
| クァ(kwa) | クィ(*kwi*) | | クェ(*kwe*) | クォ(*kwo*) |
| | ツィ(*tsi*) | | | |
| | | トゥ(*tu*) | | |
| グァ(gwa) | | | | |
| | | ドゥ(*du*) | | |
| ヴァ(*va*) | ヴィ(*vi*) | ヴ (*v*) | ヴェ(*ve*) | ヴォ(*vo*) |
| | | テュ(*tyu*) | | |
| | | フュ(*fyu*) | | |
| | | ヴュ(*vyu*) | | |

Note: No official romanized spellings for these elements except "クァ(kwa)" and "グァ(gwa)".

Table 2: The second table in the official guideline

However, the use of these elements is common and stable, because these sounds can be composed by Japanese consonants and vowels; i.e., only the combinations are new.

Each element in the second table is used to describe a non-Japanese sound that includes a non-Japanese consonant or vowel. Therefore, the use of these elements are uncommon and unstable. For example, the sound "v" in "violin" is not a Japanese sound. According to the second table, it is spelled as "ヴァ(va)". However, it is more frequently spelled as "バ (ba)", which corresponds to a Japanese sound, defined in Table-1A. This phenomenon is called *sound approximation*, which is often observed in transliteration. As a result, two transliterations of "violin", "ヴァイオリン (va-i-o-ri-n)" and "バイオリン (ba-i-o-ri-n)", exist in Japanese and both satisfy the official guideline. Note that these two spellings are the same in pronunciation; i.e., Japanese people do not distinguish "ヴァ(va)" and "バ (ba)" in pronunciation.



Note: The element "シァ(shA)" is not defined in the official guideline.

Figure 1: Variant production of "initiative"

### 2.2. Variant Production

This type of variant production of Katakana loanwords can be modeled by combination of different local mappings in transliteration. Figure 1 shows a model of transliteration of "initiative". This model produces 32 ($= 1 \times 1 \times 4 \times 4 \times 2$) different transliterations, and only a half of them are registered in UniDic-2.1.0 as mentioned above. This fact suggests that a dictionary cannot cover all possible variants; we need a dictionary look-up method that enables to retrieve a word from its unregistered variant.

## 3. Related Work

Necessity of handling Katakana variants in Japanese language processing has been recognized since 1980s and a lot of studies have been conducted up to now. These studies can be classified into four major types.

### 3.1. Dictionary Look-up

In machine translation, dictionary look-up is a crucial step to obtain information about word translation. In this step, variants cause failures because a bilingual dictionary does not cover all variants. Itsui et al. (1989) reported a method of handling Katakana variants in their machine translation system. This method uses 57 rules for variant generation in the process of the dictionary look-up but its performance is not reported clearly. Shishibori and Aoe (1993) proposed a similar method that uses 135 rules. Nomiyama (1990) proposed a different method, which uses back-transliteration to find an English entry from a Katakana word directly.

### 3.2. Variant Detection

Multiple spellings of a word in a single document should be avoided. Kubota et al. (1994) proposed a method of detecting Katakana variants in a document. Their method uses directed graphs to determine whether two Katakana strings are the variants of the same word. Shimazu et al. (1992) took a different approach, which uses variant generation rules, to detect Katakana variants.

### 3.3. Variant Generation for IR

In information retrieval, existence of word variants decreases the recall. A standard technique to improve the recall is *query expansion*, where related terms (usually synonyms) are added to the initial query. If the initial query includes Katakana loanwords, their variants should be added in query expansion. Kubomura and Kameda (2003) proposed a Katakana-variant generator, which uses 206 variant generation rules for this purpose. Hattori et al. (2009) proposed another generator, which uses backward-forward transliteration.

### 3.4. Variant Mining

Automatic collection of Katakana variants is helpful for any applications, because there is no exhaustive list of Katakana variants. Masuyama et al. (2004) proposed a three-step method of automatic construction of a Katakana-variant list from a large corpus, which uses context similarity to verify whether two spellings are the variants of the same word in the last step.

Assessment of effectiveness of the above methods is difficult because neither an exhaustive list of Katakana variants nor an standard test set of Katakana variants exist. To make matters worse, there is no clear rules that distinguish between a canonical form (spelling) and its variants; the guideline provides nothing about canonical forms. A newly complied dictionary UniDic-2.1.0 provides a new foundation of processing Katakana variants, because of its structure and compilation manual.

## 4. The UniDic Dictionary

In Japanese linguistics, word variants are classified into two sub-classes: *form variant* and *spelling variant* (The National Language Research Institute, 1983). This classification assumes a certain structure of a word or word family in a dictionary. The UniDic-2.1.0 is the first dictionary that has the structure.

An entry in UniDic-2.1.0 is a three-layered object, shown in Table 3. The top layer (i.e., the entry) is called *lemma*, which corresponds to a *generalized* word or a word family in a usual sense. In case of Katakana loanword, a lemma covers all transliteration variants produced from an original foreign word. A lemma consists of one or more *forms*; a form corresponds to a word in a usual sense, which has a unique pronunciation. For every form, one or more *spellings* are registered. In addition, for every lemma and form, a *representative (canonical) spelling* is defined. For example, in Table 3, the lemma consists of eight forms; 16 different spellings are registered in total.

According to this structure, word variants of Katakana loanwords are classified into two sub-classes. When two spellings belong to the same form, we call them *spelling variants*. When two spellings do not belong to the same form but belong to the same lemma, we call them *form variants*. Note that, in case of Katakana loanwords, form variants correspond to *pronunciation variants*, because there is no grammatical difference between variants.

To make this classification clear, we introduce a *path representation* of ID to every lemma, form, and spelling, such as 013717 (lemma), 013717.1 (form), and 013717.1.1 (spelling). By using this representation, classification is obvious. For example, “イニシアチブ (013717.1.1)” and “イニシアチヴ (013717.1.2)” are spelling variants of a form, because they share the form ID 013717.1. Two spellings “イニシアチヴ (013717.1.2)” and “イニシアティブ (013717.3.1)” are form variants of the lemma 013717.

UniDic-2.1.0 is a handcrafted dictionary developed by National Institute of Japanese Language and Linguistics, and there is a specification manual for human compilers (Ogura et al., 2010) that describes

1. the rules to distinguish spelling variants from form variants,
2. the rules to distinguish form variants from different lemmas, and
3. the rules to determine a representative spelling for a form and lemma.

This manual provides the first precise definition of classification of word variants in Japanese, and UniDic-2.1.0 provides its classification result. This is the reason why UniDic-2.1.0 provides a new foundation of word variants in Japanese.

## 5. Method

### 5.1. Formal Definition

In general, the process of dictionary look-up is modeled by the following function:

$$\text{look\_up}(q, D) \quad = \quad \{e | e \in D, q \in \text{spelling}(e)\} \quad (1)$$

where $D$ is a dictionary, $q$ is a query word (spelling), $e$ is an entry of $D$, and $\text{spelling}(e)$ is a function that returns all *registered* spellings of the entry $e$.

When this $\text{look\_up}$ function returns no entry for $q$, two possibilities remain:

1. $q$ is an unregistered spelling of *registered* entry, or
2. $q$ is a spelling of *unregistered* entry.

In the former case, the entry can be identified if a retrieval program has a variant generator. This extension is modeled by the following function:

$$\begin{aligned} &\widehat{\text{look\_up}}(q, D) \\ &\quad = \quad \{e | e \in D, s \in \text{spelling}(e), q \in \text{variant}(s)\} \quad (2) \end{aligned}$$

where $\text{variant}(s)$ is a variant generator that produces all *theoretically-possible* variants of a spelling $s$.

Because UniDic-2.1.0 has the layered structure mentioned above, it can be interpreted as either a set of lemmas ($L$) or a set of forms ($F$). Therefore, the following function is used instead of Function (1).

$$\text{look\_up}_F(q, F) \quad = \quad \{f | f \in F, q \in \text{spelling}(f)\} \quad (3)$$

| lemma | form | spelling |
|---|---|---|
| 013717 イニシアチブ<br>(i-ni-shi-a-chi-bu) | 013717.1 イニシアチブ<br>(i-ni-shi-a-chi-bu) | 013717.1.1 イニシアチブ<br>(i-ni-shi-a-chi-bu) |
| | | 013717.1.2 イニシアチヴ<br>(i-ni-shi-a-chi-**v**) |
| | 013717.2 イニシアチーブ<br>(i-ni-shi-a-chi-**i**-bu) | 013717.2.1 イニシアチーブ<br>(i-ni-shi-a-chi-i-bu) |
| | 013717.3 イニシアティブ<br>(i-ni-shi-a-**ti**-bu) | 013717.3.1 イニシアティブ<br>(i-ni-shi-a-ti-bu) |
| | | 013717.3.2 イニシアティヴ<br>(i-ni-shi-a-ti-**v**) |
| | 013717.4 イニシアティーブ<br>(i-ni-shi-a-**ti**-**i**-bu) | 013717.4.1 イニシアティーブ<br>(i-ni-shi-a-ti-i-bu) |
| | | 013717.4.2 イニシアティーヴ<br>(i-ni-shi-a-ti-i-**v**) |
| | 013717.5 イニシャチブ<br>(i-ni-**sha**-chi-bu) | 013717.5.1 イニシャチブ<br>(i-ni-sha-chi-bu) |
| | | 013717.5.2 イニシャチヴ<br>(i-ni-sha-chi-**v**) |
| | 013717.6 イニシャチーブ<br>(i-ni-**sha**-chi-**i**-bu) | 013717.6.1 イニシャチーブ<br>(i-ni-sha-chi-i-bu) |
| | | 013717.6.2 イニシャチーヴ<br>(i-ni-sha-chi-i-**v**) |
| | 013717.7 イニシャティブ<br>(i-ni-**sha**-**ti**-bu) | 013717.7.1 イニシァティブ<br>(i-ni-**shA**-ti-bu) |
| | | 013717.7.2 イニシァティヴ<br>(i-ni-**shA**-ti-**v**) |
| | | 013717.7.3 イニシャティブ<br>(i-ni-sha-ti-bu) |
| | | 013717.7.4 イニシャティヴ<br>(i-ni-sha-ti-**v**) |
| | 013717.8 イニチアティブ<br>(i-ni-**chi-a**-**ti**-bu) | 013717.8.1 イニチアティブ<br>(i-ni-chi-a-ti-bu) |

Note: bold characters indicate the difference from the upper representative.

Table 3: Structure of an entry in UniDic

Instead of Function (2), two functions can be defined:

$$\begin{aligned} &\widehat{\text{look\_up}}_F(q, F) \\ &\quad = \{f | f \in F, s \in \text{spelling}(f), q \in \text{variant}_S(s)\} \quad (4) \\ &\widehat{\text{look\_up}}_L(q, L) \\ &\quad = \{l | l \in L, s \in \text{spelling}(l), q \in \text{variant}_F(s)\} \quad (5) \end{aligned}$$

where $\text{variant}_S(s)$ is a spelling-variant generator, and $\text{variant}_F(s)$ is a form-variant generator. By calculating three functions (3)–(5) in sequence, for a given query spelling $q$, the following four cases are identified.

1. *registered spelling*
   If Function (3) returns a non-empty set, $q$ is a registered spelling of the obtained form $f$.
2. *spelling variant*
   If Function (3) returns the empty set and Function (4) returns a non-empty set, $q$ is an unregistered spelling of the obtained form $f$.
3. *form variant*
   If both Function (3) and (4) return the empty set and the Function (5) returns a non-empty set, $q$ is a spelling of an unregistered form of the obtained lemma $l$.
4. *out of vocabulary*
   If all of three functions return the empty set, $q$ is a spelling of an unregistered lemma.

These cases are summarized in Table 4.

The essence of the last two functions (4) and (5) is finding a registered spelling $s$, for a given spelling $q$. Therefore, we use the following functions in practice, instead of (4) and (5):

$$\begin{aligned} \text{find\_spelling}_S(q, D) &= S(D) \cap \text{variant}_S(q) \quad (6) \\ \text{find\_spelling}_F(q, D) &= S(D) \cap \text{variant}_F(q) \quad (7) \end{aligned}$$

where $S(D)$ is the set of all registered spellings in $D$, which is defined as $S(D) = \{s | e \in D, s \in \mathit{spelling}(e)\}$. Note that, in Function (6) and (7), variants are generated from a given spelling $q$, not a registered spelling.

### 5.2. Implementation

For calculating Function (6) and (7), we use the framework of *non-productive machine transliteration* (NPMT) (Sato, 2010), because variant generation can be modeled as transliteration within the same alphabet. In this framework, a variant generation rule is a simple bidirectional replacement between two substrings, such as "ヴァ(va) ↔ バ (ba)" and "シア (shi-a) ↔ シャ(sha)". For example, by applying the first rule to a spelling "$\underline{\text{ヴァ}}$イオリン (va-i-o-ri-n; violin)", a variant "$\underline{\text{バ}}$イオリン (ba-i-o-ri-n; violin)" is obtained.

| Function | | | lemma | form | spelling | note |
|---|---|---|---|---|---|---|
| (3) | (4) | (5) | | | | |
| non-empty | – | – | identified | identified | identified | registered spelling |
| empty | non-empty | – | identified | identified | unregistered | spelling variant |
| empty | empty | non-empty | identified | unregistered | unregistered | form variant |
| empty | empty | empty | unregistered | unregistered | unregistered | out of vocabulary |

Table 4: Result of dictionary look-up

| layer | variant type | # of rules | examples |
|---|---|---|---|
| 1 | spelling | 1 | ・ ↔ $\epsilon$ |
| 2 | spelling | 24 | ァ(A) ↔ ア (a), a/ー ↔ ア (a) |
| 3 | spelling | 71 | ヴァ(va) ↔ ブ (bu), テゥ(te-U) ↔ チュ(chu) |
| 4 | spelling | 5 | a/ァ(a/A) ↔ $\epsilon$ |
| 5 | form (spelling) | 1 | ー ↔ $\epsilon$ |
| 6 | form | 91 | クァ(kwa) ↔ カ (ka), シェ(she) ↔ セ (se) |
| 7 | form | 23 | アイ (a-i) ↔ イ (i), ルア (ru-a) ↔ ラ (ra) |
| total | | 216 | |

(note: $\epsilon$ means the empty string; 'a/' means that the previous vowel is "a")

Table 5: Variant generation rules

For a given query spelling, the NPMT algorithm, which uses prefix-filtering and dynamic programming, efficiently finds its variants registered in a dictionary, where variant generation and entry retrieval are executed simultaneously to reduce the search space. The detail of the algorithm is described in (Sato, 2010) and (Sato and Okada, 2011).

### 5.3. Variant Generation Rules

Variant generation rules are the heart of variant recognition. We have developed 216 rules in total according to the specification manual (Ogura et al., 2010), the official guideline (Cabinet notifications and directives, 1991), and other sources. These rules are organized in seven layers shown in Table 5. The first four layers (101 rules) are for spelling variants and the others (115 rules) are for form variants. The rules in lower layer correspond to *smaller* differences; they are applied with higher priority in order to identify the most *similar* registered spelling.

The first layer has only one rule that ignores the center dot, which is used as a word separator in Katakana loanwords. For example, a French term "café au lait" is transliterated into "カフェオレ" or "カフェ・オ・レ"; this difference is similar to the difference between "loanword" and "loan word" in English.

The second layer consists of 24 rules in total; 11 rules are related to the long vowel symbol "ー" and 13 rules are related to small characters. The pronunciation of the long vowel symbol is the same as the vowel just before the symbol. For example, the pronunciation of "ゴール (go-o-ru; goal)" is the same as that of "ゴオル (go-o-ru; goal)". The small characters are usually used in the fixed sequences defined in Table 1 and 2. However, they are *temporally* used as substitutions of the normal characters. For example, "スイーツ (su-i-i-tsu; sweets)" is sometimes written as "スィーツ (su-I-i-tsu; sweets)". There is no difference in pronunciation between two spellings.

The third layer consists of 71 rules, which bridge different spellings that have the same pronunciation. Some rules correspond to mapping between spellings according to Table 2 and spellings according to Table 1, such as ヴァ(va) ↔ バ (ba). Other rules cover mapping between spellings beyond the official guideline and spellings according to the guideline, such as テゥ(te-U) ↔ チュ(chu).

The fourth layer has five rules; each rule ignores a temporally-inserted small vowel character just after the same vowel. For example, the difference between "デジィタル (de-ji-I-ta-ru; digital)" and "デジタル (de-ji-ta-ru; digital)" is bridged by a rule in this layer.

The fifth layer has only one rule that ignores the long vowel symbol. Insertion or deletion of the long vowel symbol changes the pronunciation of the spelling, because the pronunciation of the long vowel symbol is the same as the vowel just before the symbol. Therefore, this type of variant, such as "コンピューター (ko-n-pyu-u-ta-a; computer)" and "コンピュータ (ko-n-pyu-u-ta; computer)", is classified into form (pronunciation) variant. An exception is temporal insertion of the long vowel symbol. For example, "ピーンチ (pi-i-n-chi; pinch)", which is produced from "ピンチ (pi-n-chi; pinch)" by temporal insertion of the long vowel symbol, is regarded as a spelling variant of the original spelling. This is similar to the production of "cooool" from "cool" in English.

The sixth layer has 91 rules, which bridge different spellings that have the similar pronunciation. Some rules correspond to mapping between spellings according to Table 2 and spellings according to Table 1, such as クァ(kwa) ↔ カ (ka). Other rules cover mapping between spellings according to Table 1B and spellings according to Table 1A, such as シェ(she) ↔ セ (se).

The final layer has 23 rules that are related to vowel substitution, such as アイ (a-i) ↔ イ (i) and ルア (ru-a) ↔ ラ (ra). The specification manual describes a little about this type of variant but not a few examples are observed in UniDic-2.1.0.

## 6. Experiment

We have conducted experiments to confirm the effectiveness of our method. First of all, we compiled a dictionary of

|  | original | removable |  |
|---|---|---|---|
| type-L | 25,821 | – | – |
| type-F | 5,489 | 3,764 | (68.6%) |
| type-S | 2,261 | 2,261 | (100.0%) |
| F and S | 7,750 | 6,025 | (77.7%) |

Table 6: Removable spellings

|  | output |  |  |  |
|---|---|---|---|---|
|  | form variant |  | spelling variant |  |
| form variant | 3,810 | (95.1%) | 198 | (4.9%) |
| spelling variant | 20 | (1.0%) | 1,997 | (99.0%) |

Table 7: Identification of variant type

Katakana loanwords by extracting all Katakana loanwords from UniDic-2.1.0. The compiled dictionary has 26,228 lemmas, 31,826 forms, and 34,063 spellings. The number of different spellings is 33,571; some spellings belong to multiple lemmas.

Next, we introduced three types of spelling.

**type-L** A representative spelling of a lemma,
e.g., “イニシアチブ (013717/013717.1/013717.1.1)”

**type-F** Not a representative spelling of a lemma but a representative spelling of a form,
e.g., “イニシアチーヴ (013717.2/013717.2.1)”

**type-S** Not a representative spelling,
e.g., “イニシアチヴ (013717.1.2)”

In the first experiment, we have examined the spelling-variant generator, which uses 102 rules in the first five layers[1]. We have removed all type-S spellings from the dictionary and examined whether the correct form is identified for each type-S spelling. For example, for a type-S input “イニシアチヴ (i-ni-shi-a-chi-v; 013717.1.2)”, the correct form is 013717.1 and its representative spelling is “イニシアチブ (i-ni-shi-a-chi-bu; 013717/013717.1/013717.1.1)”. For 2,239 (99.0%) among 2,261 inputs, the correct forms have been identified. Note that 10 among 22 errors are caused by bugs of UniDic-2.1.0. This result shows that our spelling-variant generator is almost perfect and we can remove almost all type-S spellings from the dictionary.

In the second experiment, we have examined how many type-F spellings can be removed from the dictionary when we use the form-variant generator with 216 rules. Table 6 shows the result. In addition to all type-S spellings, 3,764 (68.6%) among 5,489 type-F spellings can be removed. In total, 6,025 (77.7%) among 7,750 multiple spellings can be removed. Note that, for every removed spelling, the expected registered spelling is correctly retrieved as a result of dictionary look-up.

Because the variant-generation rules has the layered organization, our method can identify the variant type (i.e., spelling variant or form variant) between a query spelling and the retrieved registered spelling. Table 7 shows the accuracy of identification of variant type. From this table, we confirm that our method can distinguish two types of variant with high accuracy.

## 7. Conclusion

This paper proposed a method to recognize Katakana variants—a major type of word variant in Japanese—in the process of dictionary look-up. For a given spelling, the method efficiently finds its variants registered in a dictionary, by executing variant generation and entry retrieval simultaneously. We developed the seven-layered rule set (216 rules in total) for generation of Katakana variants. The experiment has shown that 77.7% of multiple spellings of Katakana loanwords in UniDic-2.1.0 are unnecessary (i.e., can be removed) when we use the proposed method.

## 8. Acknowledgements

This work was supported by JSPS KAKENHI 21300094.

## 9. References

Cabinet notifications and directives. 1991. How to spell foreign words (in Japanese).

Hiroyuki Hattori, Kazuhiro Seki, and Kuniaki Uehara. 2009. Generating diverse katakana variants via backward-forward transliteration for information retrieval. *IPSJ Transactions on Mathematical Modeling and its Applications*, 2(1):145–155.

Hiroyasu Itsui, Ryozo Kiyohara, Katsushi Suzuki, and Takashi Dasai. 1989. Processing of katakana variant notations (in Japanese). In *Proc. of the 38th National Convention of IPSJ*, pages 351–352.

Chiaki Kubomura and Hiroyuki Kameda. 2003. Information retrieval system with abilities of processing katakana-allographs (in Japanese). *The IEICE Transactions on Information and Systems (Japanese Edition)*, J86-D-II(3):418–428.

Jun’ichi Kubota, Yukie Shoda, Masahiro Kawai, Hirofumi Tamagawa, and Ryuichi Sugimura. 1994. A method of detecting KATAKANA variants in a document (in Japanese). *Transactions of Information Processing Society of Japan*, 35(12):2745–2751.

Takashi Masuyama, Satoshi Sekine, and Hiroshi Nakagawa. 2004. Automatic construction of Japanese KATAKANA variant list from large corpus. In *Proc. of COLING-2004*, pages 1214–1219.

Hiroshi Nomiyama. 1990. Handling notational variants of foreign-language-origin words (in Japanese). In *Proc. of the 41th National Convention of IPSJ*, volume 3, pages 191–192.

Hideki Ogura, Hanae Koiso, Yumi Fujiike, Sayaka Ikeuchi, and Yutaka Hara. 2010. Specifications of morphological information for the balanced corpus of contemporary written Japanese (in Japanese). Technical Report LR-CCG-09-02, National Institute of Japanese Language and Linguistics.

Satoshi Sato and Masaya Okada. 2011. Japanese-English cross-language headword search of Wikipedia. In *Proc. of the 9th International Conference on Terminology and Artificial Intellingence*, pages 45–51.

[1] We use the fifth layer to detect the temporal insertion of the long vowel symbol.

Satoshi Sato. 2010. Non-productive machine transliteration. In *Proc. of RIAO-2010*, pages 16–19.

Miwako Shimazu, Yumiko Yoshimura, Hideki Hirakawa, and Shinya Amano. 1992. An autocorrection function for katakana variants (in Japanese). In *Proc. of the 44th National Convention of IPSJ*, volume 3, pages 249–250.

Masami Shishibori and Jun-ichi Aoe. 1993. A method for generation and normalization of katakana variant notations. In *IPSJ SIG Technical Reports, NL–94–5*, pages 33–40.

The National Language Research Institute. 1983. *Writing-Form Variation of Words in Contemporary Japanese (in Japanese)*. Eishu Shuppan.