# A Rule-based Morpho-semantic Analyzer of the Japanese Verb Phrases of Simple Sentences *

Yukiko Sasaki Alam

Hosei University, Department of Digital Media,
3-7-2 Kajino-cho, Koganei, Tokyo, 184-8584 Japan
sasaki@hosei.ac.jp

**Abstract.** This paper presents the design and algorithms of a morpho-semantic analyzer to parse the Japanese verb phrases of simple sentences. This parser aims to understand the whole semantics of verb phrases by parsing them into semantic units, and thus differs from existing morphological analyzers that primarily segment sentences into morpho-phonemes, labeled with the classifications. Unlike other statistically aided morphological parsers, the algorithms used are based on rules derived from linguistic analysis. The present system can identify the syntactic category of the head word of a verb phrase, and, if it is a verb, the conjugation group, even when not listed in the dictionaries of the system. This ability of the system enables quick access to the dictionary in the category of the head word. The design is object-oriented, modeling linguistic constructions and the components, and it is thus easy to grasp the structures and algorithms, enhancing scalability, maintainability and portability. The system can be embedded in a larger system, and be used when the larger system starts parsing verb phrases.

**Keywords:** Morpho-semantic analyzer, Japanese verb phrases, Morphological parser, Rule-based, Object-oriented, Parsing

## 1. Introduction

The purpose of the current morpho-semantic analyzer is to understand the semantic structures of Japanese verb phrases, and to contribute to the semantic understandings of sentences. At present the system understands the morpho-semantic structures of the verb phrases of simple sentences. It employs the algorithms based on linguistic analyses and several lists of carefully grouped verbs. Being able to understand verb phrases semantically, it differs from other Japanese morphological analyzers, notably *Juman* (Kurohashi and Nagao, 2003) and *Chasen* (Matsumoto et al., 2000), which focus primarily on the segmentation of sentences into morpho-phonemes such as prefixes, suffixes, inflections, Case particles and the components of compound words, and on labeling them with syntactic classifications. Another major difference from other morphological analyzers is that the current system is exclusively rule-based while most analyzers, whether morphological or syntactic, are based on statistics, such as Uchimoto et al. (2003) and Murata et al. (2005), to name a few. The third difference is that the present system is able to identify the syntactic category, such as the adjective and the verb, of the head word of a verb phrase, and, if it is a verb, the conjugation group, even when it is not listed in the dictionaries of the system.

The current system, unlike finite state models, is easy to trace the problems and extend itself according to needs, because each step in the algorithms is linguistically accountable and



*22nd Pacific Asia Conference on Language, Information and Computation, pages 101–112*

intuitively comprehensible. It parses the Japanese verb phrases of simple sentences, but not those with modal and honorific expressions and those in complex sentences. It can be embedded in a larger parsing system such as a sentence phrase analyzer (Alam, 2007).

The system is encoded in Java programming language on an object-oriented design, consisting of several packages including those named *phrases*, *words*, *suffixes*, *vp_parsers*, *verb_lists* and *utils*. In the following, I will explain the components of each suffix class and the algorithms of their main methods all called *check*, with the *Causative* class in most detail, and then discuss future work, together with concluding remarks.

## 2. The Package of Suffixes

The package of suffixes is composed of *Causative*, *Passive*, *Desiderative*, *Negative* and *Inflection* classes. Each class represents a verbal suffix having the same name. The suffixes appear in verb phrases in the fixed order in which the classes are listed above. Each class has a static method named *check*, and the *check* methods are called for in that order by the controller class of the system. In this section, I will explain each class, with the *Causative* class in more detail, because the ideas underlying many of the algorithmic steps in each class are similar.

### 2.1. The *Causative* Class and the *check* Method

Among the Japanese verbal suffixes, the causative suffix, if any contained in the verb phrase, is the first that follows the stem of a verb. Thus, the parser must check first if it immediately appears after the stem, and the checking is performed by the static method of the *Causative* class entitled *check*. This method is called for from the class that embodies the main algorithm of the parser. The *check* method takes as the argument an object of the *verb_phrase* class, the components of which are to be identified in the process of parsing. It checks if the *verb_phrase* object contains an input string that matches a causative suffix, instantiates, when a matching string is found, a causative suffix object in the *Suffix* class, and registers the suffix object as a component of the verb phrase object, before returning to the system the verb phrase object with the added information.

#### *2.1.1. The Causative Suffix and Regular Verbs*

The most important function of the *check* method is to check if the initial portion of the unprocessed string of the verb phrase instance matches a string that forms a causative suffix. For that purpose, the class provides two sets of patterns which are in the form of *regular expressions*, one to identify a causative suffix following a consonant verb[1], and the other for a vowel verb, as illustrated below:

(1) (a) Causative suffix pattern for a consonant verb
*static final String CAUSATIVE_CV = "[kgsmnbtwr]ase"*[2]*;*
(b) Causative suffix pattern for a vowel verb
*static final String CAUSATIVE_VV = "[kgsmnbtwr][ei]sase";*

Given the two sets of causative suffix patterns, the parser is able to find out if the head word of the verb phrase is a consonant verb or a vowel verb. As shown in (1a) and (1b), the instances in the *String* class used for matching are in the form of regular expressions.

Using the two string matching sets, the stem of the head verb of the verb phrase is determined in the following way. When one of the string sequences in the two sets matches the initial portion of the unprocessed string of the verb phrase, that means that a causative suffix is found, and that the stem of the head verb is ready to be determined. When the matched string is among the set for a consonant verb, the stem should be the one of a consonant verb, and it is

[1] Japanese verbs can be divided into two major groups: consonant verbs and vowel verbs, depending upon the final sounds of the stems. The stems of consonant verbs end in consonants, while those of vowel verbs end in either of the two vowels /e/ and /i/. This classification is due to Bloch, 1946.

[2] In the formalism of *regular expressions*, for instance, in (1a) any one letter in square brackets is a candidate that can precede the string "ase", which is the causative suffix for a consonant verb.

determined to be a combination of both the passed-on string that had been processed[3] and the initial consonant of the matched string in the set. For instance, when the argument verb phrase has as its attribute representing the processed string the String object "か" (pronounced as /ka/) and the matched string is "kase" (a string in the set for a causative verb in (1a)), the stem of the head verb is computed to be "かk", a combination of "か" and the initial consonant "k" of the matched string "kase", and the dictionary form of the verb is determined to be "かく" (/kaku/)[4], created by combining the stem "かk" and "u", the non-past inflection for a consonant verb.

When a match is found, on the other hand, among the set for a vowel verb, the stem of the vowel verb is computed to be a combination of both the preprocessed string and the initial one-*hiragana* or syllable equivalent portion of the matched string[5]. Specifically, the stem of the vowel verb is a combination of the preprocessed string and the first one *hiragana* character consisting of a consonant followed by "e" or "i" or the first one *hiragana* character representing one of the two vowels /e/ or /i/. For instance, the stem "たべ" (pronounced as /tabe/ and meaning eating) is a combination of the preprocessed string "た" (/ta/) and the string "べ" (/be/), which is composed of the initial consonant letter "b" and the next vowel letter "e" in the set for a vowel verb given in (1b).

### *2.1.2. Two Pitfalls in the Process of Checking for a Causative Suffix*

In the process of identification of the causative suffix following a consonant verb, there occur two types of pitfalls, if no measures are taken to prevent them. One potential pitfall is caused by vowel verbs with one-*hiragana*- (or one Japanese syllable-) long stems such as "み" (/mi/) meaning seeing and "え" (/e/) meaning getting. A sequence of the one-*hiragana* stem of a vowel verb, for instance, "mi", followed by the string "sase" (the causative suffix for a vowel verb) is falsely parsed as a sequence of the preprocessed portion "mi" followed by "s" (the stem-final consonant of a consonant verb followed by "ase" (the causative suffix for a consonant verb). In this system, the parser always starts reading one *hiragana* character by assuming that the very initial *hiragana* character or syllable of a verb phrase cannot be part of a suffix or an inflectional ending, because it should be part of the stem of the head verb or part of the word in the other category which is the head of the verb phrase. Under the same assumption, the matching pattern in (1b) begins with one of the stem-final *hiragana* characters or syllables of vowel verbs to find the causative suffix following the vowel verb, whereas the pattern in (1a) begins with one of the stem-final consonants of consonant verbs to find the causative suffix following a consonant verb. Once the stem-final *hiragana* of a vowel verb with a one-*hiragana*-long stem has been preprocessed, the sequence cannot be found in the pattern in (1b), and the following remaining causative suffix "sase" for a vowel verb is wrongly interpreted as the stem-final consonant "s", followed by "ase", the causative suffix for a consonant verb. This does not happen when the stem of a vowel verb is more than one *hiragana* as in "たべ" (/tabe/ meaning eating) and "おき" (/oki/ meaning getting up), because the stem-final *hiragana* or syllabue, for instance "べ" (/be/) and "き" (/ki/), can be the initial portion of the matching strings for the causative suffix for vowel verbs.

---

[3] The parser, before starting any suffix checking, always reads one *hiragana* character, because the verb phrase minimally must consist of the uninflected (or stem) portion and the inflected portion.

[4] This verb means writing.

[5] Phonologically, one *hiragana* character, which is composed of either a combination of a consonant followed by a vowel or a single vowel, represents a Japanese syllable.

The solution to avoiding this problem is to find out if the verb in question is a vowel verb with a one-*hiragana*-long stem by looking it up in the list of such short vowel verbs, and, if it is, to register it as a vowel verb, but otherwise to register it as a consonant verb with the stem-final consonant "s". This checking should not take time because the list of such verbs is very short, with nine registered at present. We call this problem the *one-hiragana-stem-vowel-verb pitfall*, because similar problems are caused in other suffix classes as well.

There is another trap found in the process of assigning the head word of the verb phrase as a consonant verb. Like the case discussed above, this solution also requires the use of a list of verbs, but unlike the previous one, this problem is specific to the Causative class. Given the causative suffix matching set for consonant verbs illustrated in (1a), a certain group of consonant verbs in their potential forms causes a false interpretation as the causative forms. The stems of the consonant verbs in this group all end in a consonant followed by "as", as exemplified in "ぼかs" (/bok**as**/ meaning shading off a color). Since the verbal suffix indicating potential is the vowel /e/ for consonant verbs, the potential forms of the verbs in this group generate a sequence of a consonant followed by "as+e", for instance, "kase", which is exactly the same sequence that can be found in the causative suffix matching set for consonant verbs. The parser must distinguish between the two strings of the same letter sequence, one created from C (consonant) + /as/ (stem-final string) + /e/ (potential), and the other resulting from C (stem-final consonant) + /ase/ (causative suffix for a consonant verb). To distinguish between the potential form of an "as"-ending consonant verb and the causative suffix for a consonant verb, the parser constructs the dictionary form of the verb in question[6], looks for it in the list of "as"-ending consonant verbs, and, when found, assigns the head verb as a consonant verb while registering the suffix as potential.

### *2.1.3. Shorter Causative Suffix and the Treatment in the Current and Other Parsers*

A word of caution is in order about preparing the list of verbs with the stems ending in "as", which has been discussed in the previous section. Not all such verbs belong in the same group. In fact, verbs ending in "as" are controversial, because they can be divided semantically into two subgroups: those with causative sense that have the corresponding (non-causative) intransitive verbs, and those without any implication of causation and without the counterparts. For instance, among verbs ending in "as", the verb "活かs" (/ikas/ with the meaning of utilizing) does not imply any causation and does not have the corresponding intransive verb, while "泣かs" (/nakas/ with the meaning of causing someone to cry ) does have the non-causative intransitive counterpart of "泣k" (/nak/ with the meaning of weeping). In the latter case, the parser should be able to identify the composition of the sequence of "nak" (meaning weeping) followed by "as" (the shorter causative suffix[7] for a consonant verb). Otherwise, the dictionary must contain both verbs independently when the two verbs are semantically related because one is derived from the other. Treating such related verbs individually would lead to the larger size of the dictionary.

With respect to verbs with the "as"-ending stems, an inconsistent treatment is observed in *Chasen*[8], a large-scale Japanese morphological analyzer that parses sentences (Matsumoto et al

[6] The dictionary form is generated from (a) the preprocessed string followed by (b) the string from the set followed by (c) the non-past form "u".

[7] The shorter causative suffix differs from the longer one presented in the causative suffix matching set for consonant verbs in (1a). It is kind of a lexicalized verbal suffix, and more restricted in use in that it only affixes to a consonant verb, and not as productive as the longer one, which can be affixed to both consonant and vowel verbs to create the causative forms.

[8] Having made an enormous contribution to the fields related to natural language processing, *Chasen* is used in this paper as a reference point.

2001). For instance, "書かした" (/kakasita/ with the meaning of having caused someone to write) is a combination of /kak/ meaning writing, /as/ indicating causation, and /ta/ representing the past inflection, but it is incorrectly parsed as illustrated below:

| Surface form | Basic form |
|---|---|
| *書か* (/kaka/) | 書く (/kaku/ 'write')[9] |
| *し* (/si/) | する (/suru/ 'do') |
| *た* (/ta/) | た (/ta/ auxiliary) |

The meaning of causation contained by this string is not regognized in this analysis, and it would be difficult to obtain the real meaning of "書かした" (/kakasita/ with the meaning of having caused someone to write) from this analysis.

On the other hand, a similar string, "泣かした" (/nakasita/) should be analyzed as a combination of /nak/ (meaning weeping), /as/ (denoting causation), and /ta/ (the past inflection), but it is treated as a single verb, as given below:

| Surface form | Basic form |
|---|---|
| *泣かし* (/nakasi/) | 泣かす (/nakasu/) |
| *た* (/ta/) | た (/ta/ auxiliary) |

These analyses suggest that *Chasen* is not equipped with treading such derived verbs, and lists them as single verbs in the dictionary, resulting in the larger size of the dictionary. The size of the dictionary would increase even more if the meanings of verbs are furnished with. The present system is able to identify a derived verb containing the shorter causative suffix by examining if the form without the suffix is found in the list of consonant verbs. When found, it is a derived verb containing the meaning of causation. This checking takes place not in the *Causative* class, but at the very end of the parsing for double checking by using the *Verb* class.

The above is a case in which the shorter causative was not parsed as such or wrongly parsed. There is another type of case involving verbs in the same group that would require careful handling, but is not properly dealt with. The potential form of an "as"-ending verb that does not have the implication of causation does not seem to be analyzed as such. "活かせた" (/ikaseta/ with the meaning of having being able to utilize) is a combination of /ikas/ meaning utilizing, /e/ indicating potential, and /ta/ representing the past inflection, but is analyzed as follows:

| Surface form | Basic form |
|---|---|
| *活* (/katu/) | 活 (/katu/ NOUN) |
| *かせ* (/kase/) | かせる (/kaseru/ )[10] |
| *た* (/ta/) | た (/ta/ auxiliary) |

The part of speech of the head verb is analyzed as a noun, and thus the pronunciation is wrong with the one used for the noun, not for the verb. This suggests that the dictionary does not list the verb "活かす", thus resulting in a wrong analysis. The proposed model is able to parse most verbs correctly even when they are not listed in the dictionaries, because the decision relies on the forms of verbal suffixes. It is able to parse such verbs as "活かす" (/ikasu/ meaning utilizing) correctly even when they are not listed in the dictionaries.

[9] The meanings are inserted by the author because *Chasen* does not provide the meanings of verbs nor the functions of verbal suffixes. The pronunciations are transcribed there in Japanese orthography, but converted into roman letters for convenience sake in this paper.

[10] The verb "かせる" (/kaseru/) is not a modern word, and not listed in a modern Japanese dictionary.

### *2.1.4. The Irregular Verbs*

The above treatments are for regular verbs, but Japanese has two irregular verbs, which require different treatments. Their non-past (dictionary) forms are "する" (/suru/) and "くる" (/kuru/), respectively meaning doing and coming. The causative form of "する" (/suru/) is "させる" (/saseru/)[11], and thus the string sequence of "se" that follows the preprocessed string "sa" must be identified as the causative suffix for this particular irregular verb. The minimum length required for parsing this string is four letters, because a shortest remaining string could be a combination of "se" followed by an inflection of a shortest string such as the past form "ta" and the non-past form "ru".

The stem of the other irregular verb is "来" (/ku/) when written in Chinese character or "く" (/ku/) in *hiragana*. The causative form of this verb is "来させ" (/kosase/) or "こさせ" (/kosase/). As the parser always reads the first character, "来" or "こ" in this instance, it must recognize the sequence of "させ" (/sase/) together with the preprocessed string "来" or "こ" (/ko/).

### *2.1.5. The Algorithm used in the check Method*

Figure 1 in the Appendix illustrates the algorithm used in the *check* method in the *Causative* class. The algorithm proceeds in the descending order of maximum string length required to examine. As the checking for a sequence of a vowel verb followed by the causative suffix requires the longest string, it is performed first, and if the sequence looked for is found, the vowel verb is registered as the head word of the verb phrase instance, together with the causative suffix instance. Once the head word and the causative suffix have been registered, the other steps that follow are skipped.

## 2.2. The *Passive* Class and the *check* Method

A search for the passive suffix takes place in a similar way to that for the causative suffix. The parser examines if the initial letter sequence of the unprocessed string matches the string that forms the passive suffix. Similarly, two matching sets of strings are provided with: one to look for the passive suffix following a consonant verb, and the other for the passive suffix following a vowel verb, as below:

(2) (a) *static final String PASSIVE_CV = "[kgsmnbtwr]are";*
(b) *static final String PASSIVE_VV = "[kgsmnbtwr][ei]rare";*

As (2a) suggests, the identification of the passive suffix after a consonant verb is made by checking if the stem-final consonant of a consonant verb is followed by the passive suffix "are". As (2b) shows, on the other hand, the identification of the passive suffix after a vowel verb is made by examining if the stem-final syllable, for instance, "ke" or "ki" is followed by the passive suffix "rare" for a vowel verb[12].

The big difference in the looking for the causative and passive suffixes is that the passive suffix can follow either a verb or the causative suffix, whereas there is no suffix other than a

[11] The sound of the stem of this verb also changes according to a verbal suffix or inflection that follows. The stem "す" (/su/) of the irregular verb "する" (/suru/) in the non-past dictionary form changes to "さ" (/sa/) before the causative suffix, resulting in "させる" (/saseru/ meaning causing someone to do).

[12] In fact, the stem-final syllable of the stem of a vowel verb does not always consist of a consonant followed by a vowel, such as /ke/ and /ki/, but can consist of a single vowel such as /e/ and /i/ because a vowel can represent a Japanese syllable. However, as the length of such stem-final syllables is shorter by one letter than those consisting of a consonant and a vowel, and the algorithm of the system uses the length of the string for checking, they cannot be included in the pattern given in (2b).

verb that precedes the causative suffix. Therefore, the algorithm in the *check* method of the *Passive* class starts with a yes-no question. It begins by asking if the head word (or verb) has already been registered in the verb phrase instance, while there is no need for such a question in the search for the causative suffix. In the looking for a passive suffix, when the head verb has already been registered, then the decision about the verb group can be dispensed with, and the system skipps the verb group-finding procedure described in the previous paragraph, and only checks if the passive suffix follows the causative suffix. When the causative suffix precedes, since it conjugates like a vowel verb, the system examines if the unprocessed string passed on begins with “rare”, the passive suffix for a vowel verb, and if it does, an instance of the *Passive* class is created, and registered in the verb phrase instance.

There is a similar problem in the process of checking for the passive suffix to that for the causative suffix, which has previously been termed the *one-hiragana-stem-vowel-verb pitfall*. The cause of the problem is that as the entire stem of such a vowel verb is always preprocessed, and the passive suffix “rare” that follows is falsely interpreted as a stem-final consonant (“r” in this case) followed by the passive suffix “are” for a consonant verb. As in the checking of the causative suffix, the parser must resort to the use of the same very short list of such vowel verbs to avoid the misinterpretation of a vowel verb as a consonant verb.

## 2.3. The *Desiderative* Class

The desiderative suffix “た” (/ta/), which means wanting, is transcribed in one *hiragana*. Unlike the causative and passive suffixes, which conjugate like a vowel verb, the desiderative suffix conjugates like an adjective. Table 1 shows that the desiderative suffix precedes the non-past inflection “i”, the adverbial ending “ku” [13], the TE-form “kute”, and the past inflection “katta”.

**Table 1**: String sequences (consisting of the desiderative suffix “ta” followed by possible inflections) used to check if the desiderative suffix follows a causative or passive suffix or a vowel verb with one-*hiragana* long stem.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| non-past | t | a | i | | | | |
| adverbial | t | a | k | u | | | |
| TE-form | t | a | k | u | t | e | |
| past | t | a | k | a | t | t | a |

The algorithm in the *check* method of the *Desiderative* class first asks if the head word of the verb phrase instance has already been registered, and if it has, that means the causative or passive suffix has also been found. If the desiderative suffix appears after one of the two suffixes, the unprocessed string of the verb phrase begins with the desiderative suffix “ta”, followed by one of the four possible strings as listed in Table 1. Therefore, the algorithm, after finding out that the head word has been registered, examines four such possible sequences one after another until a match is found.

The above process is a simple one. The algorithm in the *check* method, however, must also deal with cases of the desiderative suffix immediately following a verb. The number of the sequences to examine is larger than when it follows another suffix that conjugates like a vowel verb, because the algorithm has to examine the varied contexts that precede the desiderative suffix. The desiderative suffix requires a particular context when it follows a consonant verb: it inserts the vowel ”i” after the stem-final consonant of a consonant verb to avoid an unwelcome double consonant sequence that otherwise results from the stem-final consonant of a consonant verb followed by the initial consonant of the desiderative suffix. This insertion of ”i” does not happen to a vowel verb, because the stem ends with a vowel. Possible string sequences to

[13] For instance, the negative suffix “na” requires the adverbial form of the desiderative, resulting in a sequence of “ta+ku+na+i” (“want-not-non-past” meaning not wanting to do something).

examine for the desiderative suffix immediately following a verb are listed in Table 2, in which the letter “C” stands for a consonant.

The larger circle in the first row of Table 2 denotes a sequence of a consonant followed by “i” or “e” followed by “tai”, whereas the smaller circle, a sequence of a vowel “i” or “e” followed by “tai”. The sequence in the larger circle handles (i-ii) a vowel verb with the stem ending in one *hiragana* or one syllable that consists of a C followed by “i” or “e”, for instance, “たべ+た+い” (/tabe+ta+i/ meaning wanting to eat) or “おり+た+い” (/ori+ta+i/ meaning wanting to get off). The same sequence also deals with (iii) a consonant verb with the epenthetic “i” between the stem-final consonant and “ta”, for instance, “かき+た+い” (/kak+i+ta+i/ meaning wanting to write). The sequence in the smaller circle handles (iv-v) a vowel verb with the stem ending in one *hiragana* transcribing a single vowel /i/ or /e/, for instance, “か+え+た+い” (/kae+ta+i/ meaning wanting to change). In fact, each row should contain two circles indicating such cases, resulting in 20 possible sequences to examine for the desiderative suffix immediately following a verb.

**Table 2**: String sequences consisting of the desiderative suffix ”ta” followed by possible inflections which are used to check if the desiderative suffix follows a verb (“C” stands for a consonant).

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| non-past | C | i/e | t | a | i | | | | |
| adver-bial | C | i/e | t | a | k | u | | | |
| TE-form | C | i/e | t | a | k | u | t | e | |
| past | C | i/e | t | a | k | a | t | t | a |

A problem that occurs in the process of looking for the desiderative suffix is caused by the insertion of “i” between the stem-final consonant of a consonant verb and the initial consonant of the desiderative suffix “ta”, because a consonant verb stem followed by the epenthetic “i” may result in the same sequence as a vowel verb stem ending in “i”. For instance, “おきたい” (/oki+ta+i/ with the vowel verb stem meaning getting up “oki” followed by the desiderative “ta” followed by the non-past “i”) and “かきたい” (/kak+i+ta+i/ with the consonant verb meaning writing followed by the epenthetic “i” followed by the desiderative “ta” followed by the non-past “i”) shares the same sequence of /kitai/, even though one is a vowel verb stem and the other, a consonant verb stem. After processing the initial *hiragana*, “お” (/o/) and “か” (/ka/) in these examples, the unprocessed strings are “kitai” in each example, and the only way the parser knows that one contains the epenthetic “i” is by reference to the list of vowel verbs with the stems ending in “i”. Fortunately, the number of vowel verbs with the stems ending in “i” is much smaller, with 304 such verbs listed at present. To distinguish, the parser creates the dictionary form of the verb in question, and determines whether the verb is a vowel verb by referring to the list of such vowel verbs.

Lastly, there also occurs the *one-hiragana-stem-vowel-verb pitfall* in the *check* method of the *Desiderative* class. For vowel verbs with the one-*hiragana*-long stems, string sequences in Table 1 as well as the preprocessed string must be examined, and, when a match is found and the preprocessed string is one-*hiragana* long, the dictionary form must be created and validated by looking it up in the very short list of one-*hiragana* stem vowel verbs. Two irregular verbs that precede the desiderative suffix must be treated in a similar way as in the *check* method of the *Causative* class.

## 2.4. The *Negative* Class

The Negative suffix “na” can follow immediately a verb, the causative suffix, the passive suffix or the desiderative suffix. When it immediately follows the causative or the passive suffix, the

unprocessed string of the verb phrase instance begins with “na”. Since this suffx conjugates like an adjective, the string sequences to examine is similar to those for the desiderative suffix in Table 1, as indicated in the smaller circle in Table 3.

**Table 3**: String sequences consisting of the negative suffix “na” followed by possible inflections that are used to check if the negative suffix follows a causative or passive suffix or a vowel verb with the one-*hiragana* long stem.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| non-past | k | u | n | a | i | | | | |
| adverbial | k | u | n | a | k | u | | | |
| TE-form | k | u | n | a | k | u | t | e | |
| past | k | u | n | a | k | a | t | t | a |

The difference from Table 1 for the desiderative suffix is that Table 3 has an extra set of the string “ku” preceding the negative suffix. Although omitted, the larger and smaller circles must be on each row of Table 3 as on the first row. The sequences in the larger circles are used to identify the sequence of the negative suffix immediately following an adjective or the suffix that conjugates like an adjective such as the desiderative. The negative suffix requires the adverbial inflection “ku” for an adjective or the equivalent in conjugation.

When the head verb (or word) has not been identified, the algorithm needs to find out the head word and its syntactic category, and, if it is a verb, its conjugation group. To identify these, the algorithm requires information on the contexts that precede the negative suffix. Unlike the vowel /i/ for the desiderative suffix, the vowel /a/ is inserted between the stem-final consonant of a consonant verb and the initial consonant of the negative suffix. Table 4 shows the vowels, /i/, /e/ and /a/ each of which can exist between a consonant and the initial consonant of the negative suffix. The former two vowels, /i/ and /e/, are the stem-final vowels of vowel verbs, and the last vowel /a/ is the epenthetic vowel used to avoid an unwelcome double consonant sequence resulting from the stem-final consonant of a consonant verb followed by the initial consonant of the negative suffix. Thus, the existence of “a” in that position implies a possible presence of a consonant verb, whereas the existence of “i” or “e” in that context, a plausible presence of a vowel verb, and the algorithm uses this information to determine the conjugation group of the verb in question.

**Table 4**: String sequences consisting of the negative suffix followed by possible inflections used when it follows a verb (“C” indicates a consonant).

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| non-past | C | a/e/i | n | a | i | | | | |
| adverbial | C | a/e/i | n | a | k | u | | | |
| TE-form | C | a/e/i | n | a | k | u | t | e | |
| past | C | a/e/i | n | a | k | a | t | t | a |

Like Table 2 for the desiderative suffix, the two circles on the first row on Table 4 indicate two sequences of strings, the longer of which deals with (i-ii) such vowel verb examples as “たべ+な+い” (/tabe+na+i/ meaning not eating) or ”おり+な+い” (/ori+na+i/ meaning not geting off). The longer circle also includes the treatment of (iii) the negative form of a consonat verb with the epenthetic vowel /a/ such as “かか+な+い” (/kak+a+na+i/ meaning not writing). The shorter circle is to handle (iv-v) such an instance as “かえ+な+い” (/kae+na+i/ meaning not changing), involving a vowel verb the stem of which ends in a vowel /i/ or /e/ transcribed in one *hiragana*. Each row having two such circles and five strings to examine, thus altogether 20 strings are provided with for checking.

As in the case of the desiderative suffix, vowel verbs with the stems of one *hiragana* followed by the negative suffix are identified by referring to sequences listed in Table 3 as well as the very short list of one-*hiragana* stem vowel verbs.

## 2.5. The *Inflection* Class

The *check* method in the *Inflection* class also begins by asking whether the head word has been found or not. When it has been found and registered, that means a suffix that may appear before the inflection has also been recorded in the verb phrase instance. The verbal suffixes either conjugate like a vowel verb as the causative and passive suffixes do or like an adjective as the desiderative and negative suffixes do[14]. Table 5 shows the string sequences used to identify the inflections for verbal suffixes that conjugate like vowel verbs or adjectives.

**Table 5**: String sequences representing the inflections to examine when the head word has been found and the inflection immediately follows a verbal suffix ("ADJ" stands for an adjective or a suffix that conjugates like an adjective, and "VV", a vowel verb or a suffix that conjugates like a vowel verb).

| | | | | | |
|---|---|---|---|---|---|
| non-past (after a VV stem) | r | u | | | |
| non-past (after an ADJ stem) | i | | | | |
| past (after a VV stem) | t | a | | | |
| past (after a ADJ stem) | k | a | t | t | a |
| adverbial (after an ADJ stem) | k | u | | | |
| TE-form (after a VV stem or an ADJ adverbial form) | t | e | | | |
| TE-form (after an ADJ stem) | k | u | t | e | |

When the head verb (or word) has not been identified yet in the process of checking for suffixes, it should be recognized by resorting to the strings listed in Table 6. A row containing "C" actually has two string sequences each, one beginning with a consonant indicated by "C" in the larger circle, and the other without it in the smaller circle.

**Table 6**: String sequences containing inflections used to identify a head verb and the conjugation group ("VV" indicates a vowel verb, and "CV", a consonant verb).

| | | | | |
|---|---|---|---|---|
| non-past (for a VV) | C | i/e | r | u |
| non-past (for a CV) | C | u[15] | | |
| non-past (for a CV with the stem-final "w" appearing in the negative form) | C | a/u/o | u | |
| past (for a VV) | C | i/e | t | a |
| past (for a CV) | i/t | t | a | |
| past (for a CV) | i/n | d | a | |
| past (for a CV) | s | i | t | a |
| TE-form (for a VV) | C | i/e | t | e |
| TE-form (for a CV) | i/t | t | e | |
| TE-form (for a CV) | i/n | d | e | |
| TE-form (for a CV) | s | i | t | e |

The Japanese consonant verbs undergo sound changes in the past forms as well as the TE-forms, and the parser must recover the stem-final consonants from the altered consonants found in the

[14] There are other suffixes that appear before inflections, and conjugate like consonant verbs. Among them are honorific suffixes such as /nasar/ and the suffix meaning appearing to want, /tagar/. The current system does not handle them. In addition, some words or modal suffixes are syntactically nouns, requiring inflections that follow nouns. When the current system is extended to handle more than simple sentences, Table 5 should include inflections for nouns as well as a certain group of consonant verbs.

[15] The second row also should have two string sequences, "C+u" and "u" alone. The string "u" is applied only after the unprocessed string has been exhaustively examined in vain. The string is used to identify such verbs as " いう" (/iu/ meaning saying) and "あう" (/au/ meaning meeting) when the unprocessed string consists of "う" (/u/) after the one-*hiragana*-long stem has been processed.

past forms or the TE-forms. Once the stems are identified, it is easy to form the dictionary (or non-past) forms, because they are combinations of the stems and the non-past form “u”.

The current system has a class entitled *Verb*, which has a static method that computes the dictionary form of a consonant verb in reference to the past form or TE-form. It generates possible dictionary forms, looks them up in the lists of consonant verbs that are grouped accoding to conjugation types, and when found, it determines the dictionary forms. For that purpose, consonant verbs are divided into four groups, based on different sound changes occurring in the past and TE-forms.

## 3. Future Work and Conclusion

The current parser handles verb phrases of simple sentences, and still remains to be extended to handle complex verb phrases containing modal, honorific and polite expressions as well as verb phrases ending in the conjunctive and conditional forms. The system is designed to be object-oriented, simulating linguistic components, and the algorithms are easy to understand intuitively, because each step is linguistically accountable. Thus it offers high scalability, maintainability and portability. The current parser is powerful in that it is able to identify the syntactic category of the head word of the verb phrase, and, if it is a verb, the conjugation group via algorithmic inference even when it is an unknown word. The current system uniquely differs from other morphological analyzers in that its aim is to segment verb phrases into semantic units to understand the semantics of the verb phrases rather than to segment sentences into morpho-phonemes and label them with syntactic classification. The present system can easily be embedded in a larger parsing system. Much work remains to be done, but the objective of the proposed system and the methodology employed seem to be promising.

## References


Alam, Sasaki Yukiko. 2007. Analyzer to Identify Phrases and the Functional Roles in Sentences: Its Architectural Aspects. *Proceedings of PACLIC 21*, pp. 67-75.

Bloch, Bernard. 1946. Studies in Colloquial Japanese–Inflection. In Roy Andrew Miller, ed., *Bernard Bloch on Japanese*, pp. 1-24. New Haven: Yale University Press.

Covington, Michael A. 1990. *A Dependency Parser for Variable Word-Order Languages. Research Report AI-1990-01*. Artificial Intelligence Programs University of Georgia.

Fuchi, Takeshi and Shinichiro Takagi. 1998. Japanese Morphological Analyzer using Word Co-occurrence. *Proceedings of the COLING*, pp. 409-413.

Kameda, Masayuki. 1996. A Portable & Quick Japanese Parser: QJP. *Proceedings of the COLING*, pp. 616-621.

Kashioka, Hideki, Yasuhiro Kawata and Yumiko Kinjo. 1998. Use of Mutual Information Based Character Clusters in Dictionary-less Morphological Analysis of Japanese. *Proceedings of the COLING*, pp. 658-662.

Kazama, Jun’ichi. 2001. *Adaptive Morphological Analysis with a Small Tagged Corpus*. Master Thesis: University of Tokyo.

Kurohashi, Sadao and Makoto Nagao. 2003. Building a Japanese Parsed corpus – while improving the parsing system. In Anne Abeille, ed., *Treebank Building Using Parsed Corpora*, pp. 249-260. Dordrecht: Kluwer Academic Publishers.

Matsumoto, Yuji, Akira Kitauchi, TatsuoYamashita, Yoshitaka Hirano, Hiroshi Matsuda, Kazuma Takaoka and Masayuki Asahara. 2001. *Morphological Analysis System ChaSen version 2.2.4 Manual*. Nara, Japan: Nara Institute of Science and Technology.

Murata, Masaki, Masao Utiyama, Hiroshi Isahara and Qing Ma. 2005. Correction of Errors in a Verb Modality Corpus for Machine Translation with a Machine-Learning Method. *ACM Transactions on Asian Language Information Processing*, 4 (1), 18-37.

Uchimoto, Kiyotaka, Chikashi Nobata, Atsushi Yamada, Satoshi Sekine and Hitoshi Isahara. 2003. Morphological Analysis of a Large Spontaneous Speech Corpus in Japanese.

*Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 479-488.

## Appendix

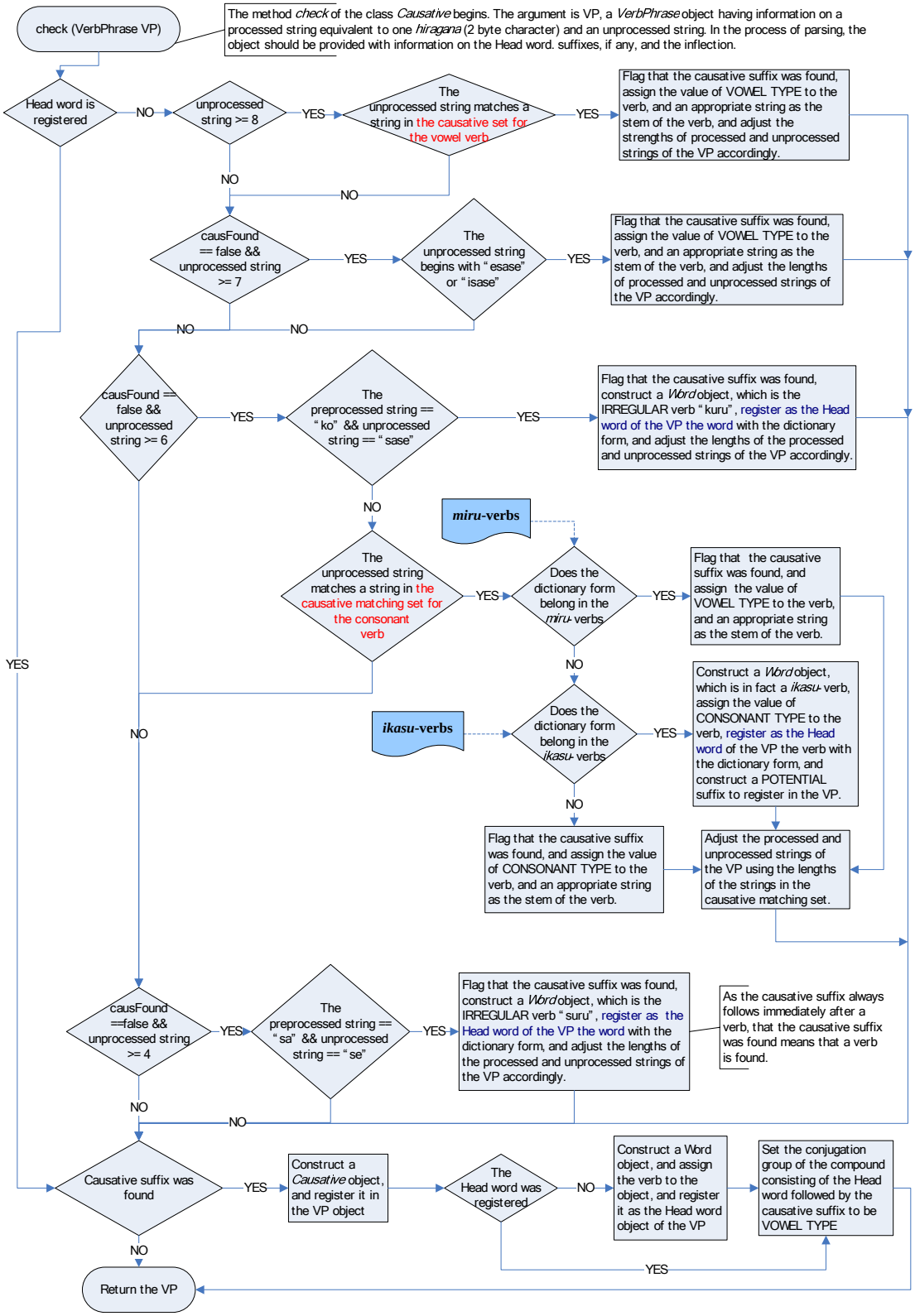


**Figure 1**: Algorithm used in the *check* method of the *Causative* class