Automatic Construction of Japanese KATAKANA Variant List from Large Corpus

Takeshi Masuyama
Information Technology Center
University of Tokyo
7-3-1, Hongo, Bunkyo
Tokyo 113-0023
Japan
tak@r.dl.itc.u-tokyo.ac.jp

Satoshi Sekine
Computer Science Department
New York University
715 Broadway, 7th floor
New York NY 10003
USA
sekine@cs.nyu.edu

Hiroshi Nakagawa
Information Technology Center
University of Tokyo
7-3-1, Hongo, Bunkyo
Tokyo 113-0023
Japan
nakagawa@dl.itc.u-tokyo.ac.jp

Abstract

This paper presents a method to construct Japanese KATAKANA variant list from large corpus. Our method is useful for information retrieval, information extraction, question answering, and so on, because KATAKANA words tend to be used as "loan words" and the transliteration causes several variations of spelling. Our method consists of three steps. At step 1, our system collects KATAKANA words from large corpus. At step 2, our system collects candidate pairs of KATAKANA variants from the collected KATAKANA words using a spelling similarity which is based on the edit distance. At step 3, our system selects variant pairs from the candidate pairs using a semantic similarity which is calculated by a vector space model of a context of each KATAKANA word. We conducted experiments using 38 years of Japanese newspaper articles and constructed Japanese KATAKANA variant list with the performance of 97.4% recall and 89.1% precision. Estimating from this precision, our system can extract 178,569 variant pairs from the corpus.

1 Introduction

"Loan words" in Japanese are usually written by a phonogram type of Japanese character set, KATAKANA. Because of loan words, the transliteration causes several variations of spelling. Therefore, Japanese KATAKANA words sometimes have several different orthographies for each original word. For example, we found at least six different spellings of "spaghetti" in 38 years of Japanese newspaper articles, such as "スパゲッティ," "スパゲッティー," "スパゲッテイ," "スパゲティ," "スパゲティー," and "スパゲテイ." The different expression causes problems when we use search engines, question answering systems, and so on (Yamamoto et al., 2003). For example, when we input "スパゲッティ" as a query for a search engine or a query for a question answering system, we may not be able to find the web pages or the answers for which we are looking, if a different orthography for "スパゲッティ" is used.

We investigated how many documents were retrieved by Google [1] when each Japanese KATAKANA variant of "spaghetti" was used as a query. The result is shown as Table 1.

For example, when we inputted "スパゲッティ" as a query of Google, 104,000 documents were retrieved and the percentage was 34.6%, calculated by 104,000 divided by 300,556. From Table 1, we see that each of six variants appears frequently and thus we may not be able to find the web pages for which we are looking.

Although we can manually create Japanese KATAKANA variant list, it is a labor-intensive task. In order to solve the problem, we propose an automatic method to construct Japanese KATAKANA variant list from large corpus.

Variant	# of retrieved documents
スパゲッティ	104,000 (34.6%)
スパゲッティー	25,400 (8.5%)
スパゲッテイ	1,570 (0.5%)
スパゲティ	131,000 (43.6%)
スパゲティー	37,700 (12.5%)
スパゲテイ	886 (0.3%)
Total	300,556 (100%)

Table 1: Number of retrieved documents when we inputted each Japanese KATAKANA variant of "spaghetti" as a query of Google.

Our method consists of three steps. First, we collect Japanese KATAKANA words from large corpus. Then, we collect candidate pairs of KATAKANA variants based on a spelling similarity from the collected Japanese KATAKANA words. Finally, we select variant pairs using

[1] http://www.google.co.jp/

a semantic similarity based on a vector space model of a context of each KATAKANA word.

This paper is organized as follows. Section 2 describes related work. Section 3 presents our method to construct Japanese KATAKANA variant list from large corpus. Section 4 shows some experimental results using 38 years of Japanese newspaper articles, which we call “the Corpus” from now on, followed by evaluation and discussion. Section 5 describes future work. Section 6 offers some concluding remarks.

2 Related Work

There are some related work for the problems with Japanese spelling variations. In (Shishibori and Aoe, 1993), they have proposed a method for generating Japanese KATAKANA variants by using replacement rules, such as ベ (be) ↔ ヴェ (ve) and チ (chi) ↔ ツィ (tsi). Here, “↔” represents “substitution.” For example, when we apply these rules to “ベネチア (Venezia),” three different spellings are generated as variants, such as “ベネツィア,” “ヴェネチア,” and “ヴェネツィア.”

Kubota et al. have extracted Japanese KATAKANA variants by first transforming KATAKANA words to directed graphs based on rewrite rules and by then checking whether the directed graphs contain the same labeled path or not (Kubota et al., 1993). A part of their rewrite rules is shown in Table 2. For example, when applying these rules to “クウェート (Kuwait),” “ク aαc,” “ク bαc,” “クウ dαc” are generated as variants.

KATAKANA String → Symbol
ウェ (we), エ (e) → a
ウェ (we), ウエ (ue) → b
トゥ (twu), ト (to), ツ (tsu) → c
ー (macron) → α
ェ (small e), エ (e) → d

Table 2: A part of rewrite rules.

In (Shishibori and Aoe, 1993) and (Kubota et al., 1993), they only paid attention to applying their replacement or rewrite rules to words themselves and didn’t pay attention to their contexts. Therefore, they wrongly decide that “ウエーブ” is a variant of “ウェブ.” Here, “ウエーブ” represents “wave” and “ウェブ” represents “web.” In our method, we will decide if “ ウエーブ” and “ウェブ” convey the same meaning or not using a semantic similarity based on their contexts.

3 Construct Japanese KATAKANA Variant List from Large Corpus

Our method consists of the following three steps.

1. Collect Japanese KATAKANA words from large corpus.
2. Collect candidate pairs of KATAKANA variants from the collected KATAKANA words using a spelling similarity.
3. Select variant pairs from the candidate pairs based on a semantic similarity.

3.1 Collect KATAKANA Words from Large Corpus

At the first step, we collected Japanese KATAKANA words which consist of a KATAKANA character, ・ (bullet), ー (macron-1), and ― (macron-2), which are commonly used as a part of KATAKANA words, using pattern matching. For example, our system collects three KATAKANA words “ ルートウィヒ・エアハルト (Ludwig Erhard-1),” “ソ (Soviet),” “ルードウィッヒ・エアハルト (Ludwig Erhard-2),” “ドイツ (Germany)” from the following sentences. Note that two mentions of “Ludwig Erhard” have different orthographies.

- “奇跡の経済復興の父” といわれる故**ルートウィヒ・エアハルト**氏。(Defunct Ludwig Erhard-1 is called “Father of The Miraculous Economic Revival.”)
- もし**ソ**連や東欧諸国が統制志向を捨て、一九四八年に西独の**ルードウィッヒ・エアハルト**がとったような経済の自由化へと突き進めば、西**ドイツ**のように奇跡の復興を遂げるかもしれない。(If Soviet and East European countries give up their controlling concepts and pursue the economic deregulation which Ludwig Erhard-2 of West Germany did in 1948, they may achieve the miraculous revival like West Germany.)

3.2 Spelling Similarity

At the second step, our system collects candidate pairs of two KATAKANA words, which are similar in spelling, from the collected KATAKANA words described in Section 3.1. We used “string penalty” to collect candidate pairs. String penalty is based on the edit distance (Hall and DOWLING, 1980) which is a similarity measure between two strings. We used the following three types of operations.

- Substitution
  Replace a character with another character.
- Deletion
  Delete a character.
- Insertion
  Insert a character.

We also added some scoring heuristics to the operations based on a pronunciation similarity between characters. The rules are tuned by hand using randomly selected training data.

Some examples are shown in Table 3. Here, "↔" represents "substitution" and lines without ↔ represent "deletion" or "insertion." Note that "Penalty" represents a score of the string penalty from now on.

For example, we give penalty 1 between "ジャーナル" and "ヂャーナル," because the strings become the same when we replace "ヂ" with "ジ" and its penalty is 1 as shown in Table 3.

Rules	Penalty
ア (a) ↔ ァ (small a)	1
ジ (zi) ↔ ヂ (di)	1
ー (macron)	1
ハ (ha) ↔ バ (ba)	2
ウ (u) ↔ ヴ (vu)	2
ア (a) ↔ ヤ (ya)	3
ツ (tsu) ↔ ッ (small tsu)	3

Table 3: A part of our string penalty rules.

We analyzed hundreds of candidate pairs of training data and figured out that most KATAKANA variations occur when the string penalties were less than a certain threshold. In this paper, we set 4 for the threshold and regard KATAKANA pairs as candidate pairs when the string penalties are less than 4. The threshold was tuned by hand using randomly selected training data.

For example, from the collected KATAKANA words described in Section 3.1, our system collects the pair of ルートウィヒ・エアハルト and ルードウィッヒ・エアハルト， since the string penalty is 3.

3.3 Context Similarity

At the final step, our system selects variant pairs from the candidate pairs described in Section 3.2 based on a semantic similarity. We used a vector space model as a semantic similarity.

In the vector space model, we treated 10 randomly selected articles from the Corpus as a context of each KATAKANA word.

We divided sentences of the articles into words using JUMAN[2] (Kurohashi and Nagao, 1999) which is the Japanese morphological analyzer, and then extracted content words which consist of nouns, verbs, adjectives, adverbs, and unknown words except stopwords. Stopwords are composed of Japanese HIRAGANA characters, punctuations, numerals, common words, and so on.

We used a cosine measure to calculate a semantic similarity of two KATAKANA words. Suppose that one KATAKANA word makes a context vector $\mathbf{a}$ and the other one makes $\mathbf{b}$. The semantic similarity between two vectors $\mathbf{a}$ and $\mathbf{b}$ is calculated as follows.

$$sim(\mathbf{a},\mathbf{b}) = cos\theta = \frac{\mathbf{a}\cdot\mathbf{b}}{|\mathbf{a}||\mathbf{b}|} \qquad (1)$$

The cosine measure tends to overscore frequently appeared words. Therefore, in order to avoid the problem, we treated $log(N+1)$ as a score of a word appeared in a context. Here, N represents the frequency of a word in a context.

We set 0.05 for the threshold of the semantic similarity, i.e. we regard candidate pairs as variant pairs when the semantic similarities are more than 0.05. The threshold was tuned by hand using randomly selected training data.

In the case of "ルートウィヒ・エアハルト (Ludwig Erhard-1)" and "ルードウィッヒ・エアハルト (Ludwig Erhard-2)", the semantic similarity becomes 0.17 as shown in Table 4. Therefore, we regard them as a variant pair.

Note that in Table 4, a decimal number represents a score of a word appeared in a context calculated by $log(N+1)$. For example, the score of 奇跡 (miracle) in the first context is 0.7.

4 Experiments

4.1 Data Preprocessing and Performance Measures

We conducted the experiments using the Corpus. The number of documents in the Cor-

[2] http://www.kc.t.u-tokyo.ac.jp/nl-resource/juman.html

Word	ルートウィヒ・エアハルト
Context	奇跡 (miracle):0.7 経済 (economy):1.9 父 (father):0.7 復興 (revival):0.7 . . .
Word	ルードウィッヒ・エアハルト
Context	奇跡 (miracle):1.1 自由化 (liberalization):1.4 経済 (economy):2.4 復興 (revival):1.1 . . .
Similarity	0.17

Table 4: Semantic similarity between "ルートウィヒ・エアハルト" and "ルードウィッヒ・エアハルト."

pus was 4,678,040 and the distinct number of KATAKANA words in the Corpus was 1,102,108.

As for a test set, we collected candidate pairs whose string penalties range from 1 to 12. The number of collected candidate pairs was 2,590,240. In order to create sample correct KATAKANA variant data, 500 out of 2,590,240 were randomly selected and we evaluated them manually by checking their contexts. Through the evaluation, we found that no correct variant pairs appeared from 10 to 12. Thus, we think that treating candidate pairs whose string penalties range from 1 to 12 can cover almost all of correct variant pairs.

To evaluate our method, we used recall (Re), precision (Pr), and F measure (F). These performance measures are calculated by the following formulas:

$$\begin{aligned} Re &= \frac{number\,of\,pairs\,found\,and\,correct}{total\,number\,of\,pairs\,correct}, \\ Pr &= \frac{number\,of\,pairs\,found\,and\,correct}{total\,number\,of\,pairs\,found}, \\ F &= \frac{2RePr}{Re + Pr}. \end{aligned}$$

4.2 Experiment-1

We conducted the first experiment based on two settings; one method uses only the spelling similarity and the other method uses both the spelling similarity and the semantic similarity.

Henceforth, we use "Method$_{\mathrm{p}}$," "Method$_{\mathrm{p\&s}}$," "Ext," and "Cor" as the following meanings.

Method$_{\mathrm{p}}$: The method using only the spelling similarity

Method$_{\mathrm{p\&s}}$: The method using both the spelling similarity and the semantic similarity

Ext: The number of extracted candidate pairs

Cor: The number of correct variant pairs among the extracted candidate pairs

Note that in Method$_{\mathrm{p\&s}}$, we ignored candidate pairs whose string penalties ranged from 4 to 12, since we set 4 for the threshold of the string penalty as described in Section 3.2.

The result is shown in Table 5. For example, when the penalty was 2, 81 out of 117 were selected as correct variant pairs in Method$_{\mathrm{p}}$ and the precision was 69.2%. Also, 80 out of 98 were selected as correct variant pairs in Method$_{\mathrm{p\&s}}$ and the precision was 81.6%.

As for Penalty 1-12 of Method$_{\mathrm{p}}$, i.e. we focused on the string penalties between 1 and 12, the recall was 100%, because we regarded 269 out of 500 as correct variant pairs and Method$_{\mathrm{p}}$ extracted all of them. Also, the precision was 53.8%, calculated by 269 divided by 500. Comparing Method$_{\mathrm{p\&s}}$ to Method$_{\mathrm{p}}$, the recall and the precision of Method$_{\mathrm{p\&s}}$ were well-balanced, since the recall was 97.4% and the precision was 89.1%.

In the same way, for Penalty 1-3, i.e. the string penalties between 1 and 3, the recall of Method$_{\mathrm{p}}$ was 98.1%, since five correct variant pairs between 4 and 12 were ignored and the remaining 264 out of 269 were found. The precision of Method$_{\mathrm{p}}$ was 77.2%. It was 23.4% higher than the one of Penalty 1-12. Thus, F measure also improved 16.4%. This result indicates that setting 4 for the threshold works well to improve overall performance.

Now, comparing Method$_{\mathrm{p\&s}}$ to Method$_{\mathrm{p}}$ when the string penalties ranged from 1 to 3, the recall of Method$_{\mathrm{p\&s}}$ was 0.7% lower. This was because Method$_{\mathrm{p\&s}}$ couldn't select two correct variant pairs when the penalties were 1 and 2. However, the precision of Method$_{\mathrm{p\&s}}$ was 16.2% higher. Thus, F measure of Method$_{\mathrm{p\&s}}$ improved 6.7% compared to the one of Method$_{\mathrm{p}}$. From this result, we think that taking the semantic similarity into account is a better strategy to construct Japanese KATAKANA variant list.

Penalty		Method$_{\mathrm{p}}$	Method$_{\mathrm{p\&s}}$
		Cor/Ext (%)	Cor/Ext (%)
1		130/134 (97.0)	129/129 (100)
2		81/117 (69.2)	80/98 (81.6)
3		53/91 (58.2)	53/67 (79.1)
4		2/14 (14.3)	
5		0/30 (0.0)	
6		1/14 (7.1)	
7		1/20 (5.0)	
8		0/14 (0.0)	
9		1/12 (8.3)	
10		0/16 (0.0)	
11		0/17 (0.0)	
12		0/21 (0.0)	
1-3	*Re*	264/269 (98.1)	262/269 (97.4)
	Pr	264/342 (77.2)	262/294 (89.1)
	F	86.4%	93.1%
1-12	*Re*	269/269 (100)	
	Pr	269/500 (53.8)	
	F	70.0%	

Table 5: Comparison of Method$_{\mathrm{p}}$ and Method$_{\mathrm{p\&s}}$.

4.3 Experiment-2

We investigated how many variant pairs were extracted in the case of six different spellings of “spaghetti” described in Section 1. Table 6 shows the result of all combination pairs when we applied Method$_{\mathrm{p\&s}}$.

For example, when the penalty was 1, Method$_{\mathrm{p\&s}}$ selected seven candidate pairs and all of them were correct. Thus, the recall was 100%. From Table 6, we see that the string penalties of all combination pairs ranged from 1 to 3 and our system selected all of them by the semantic similarity.

Penalty	Method$_{\mathrm{p\&s}}$
1	7/7 (100%)
2	6/6 (100%)
3	2/2 (100%)
Total	15/15 (100%)

Table 6: A result of six different spellings of “spaghetti” described in Section 1.

4.4 Estimation of expected correct variant pairs

We estimated how many correct variant pairs could be selected from the Corpus based on the precision of Method$_{\mathrm{p\&s}}$ as shown in Table 5. The result is shown in Table 7. We find that the number of candidate pairs in the Corpus was 100,746 for the penalty of 1, and 56,569 for the penalty of 2, and 40,004 for the penalty of 3.

For example, when the penalty was 2, we estimate that 46,178 out of 56,569 could be selected as correct variant pairs, since the precision was 81.6% as shown in Table 5. In total, we estimate that 178,569 out of 197,319 could be selected as correct variant pairs from the Corpus.

Penalty	# of expected variant pairs
1	100,746/100,746 (100%)
2	46,178/56,569 (81.6%)
3	31,645/40,004 (79.1%)
Total	178,569/197,319 (90.5%)

Table 7: Estimation of expected correct variant pairs.

4.5 Error Analysis-1

As shown in Table 5, our system couldn’t select two correct variant pairs using semantic similarity when the penalties were 1 and 2. We investigated the reason from the training data. The problem was caused because the contexts of the pairs were diffrent. For example, in the case of “アロックサンワ” and “アロック・サンワ,” which represent the same building material company “Aroc Sanwa” of Fukui prefecture in Japan, their contexts were completely different because of the following reason.

- **アロックサンワ**, **アロック・サンワ**

  アロックサンワ (Aroc Sanwa): This word appeared with the name of an athlete who took part in the national athletic meet held in Toyama prefecture in Japan, and the company sponsored the athlete.

  アロック・サンワ (Aroc・Sanwa): This word was used to introduce the company in the article.

Note that each context of these words was composed of only one article.

4.6 Error Analysis-2

From Table 5, we see that the numbers of incorrect variant pairs selected by Method$_{\mathrm{p\&s}}$ were

18 and 14 for each penalty of 2 and 3. We investigated such cases in the training data. The example of “カート (Cart, Kart)” and “カード (Card)” is shown as follows.

- **カート**, **カード**

  カート (Cart, Kart): This word was used as the abbreviation of “Shopping Cart,” “Racing Kart,” or “Sport Kart.”

  カード (Card): This word was used as the abbreviation of “Credit Card” or “Cash Card” and was also used as the meaning of “Schedule of Games.”

Although these were not a variant pair, our system regarded the pair as the variant pair, because their contexts were similar. In both contexts, “利用 (utilization),” “記録 (record),” “客 (guest),” “目指す (aim),” “チーム (team),” “優勝 (victory),” “高い (high, expensive),” “活躍 (success),” “出場 (entry),” and so on were appeared frequently and therefore the semantic similarity became high.

5 Future Work

In this paper, we have used newspaper articles to construct Japanese KATAKANA variant list. We are planning to apply our method on different types of corpus, such as patent documents and Web data. We think that more variations can be found from Web data. In the case of “spaghetti” described in Section 1, we found at least seven more different spellings, such as “スッパゲティ,” “スッパゲティー,” “スパゲーティ,” “スパゲーティー” “スパゲテー,” “スパッゲティ,” and “スパッゲティー.”

Although we have manually tuned scoring rules of the string penalty using training data, we are planning to introduce an automatic method for learning the rules.

We will also have to consider other character types of Japanese, i.e. KANJI variations and HIRAGANA variations, though we have focused on only KATAKANA variations in this paper. For example, both “引越し” and “引っこし” mean “move” in Japanese.

6 Conclusion

We have described the method to construct Japanese KATAKANA variant list from large corpus. Unlike the previous work, we focused not only on the similarity in spelling but also on the semantic similarity.

From the experiments, we found that $\text{Method}_{\text{p\&s}}$ performs better than Method_{p}, since it constructed Japanese KATAKANA variant list with high performance of 97.4% recall and 89.1% precision.

Estimating from the precision, we found that 178,569 out of 197,319 could be selected as correct variant pairs from the Corpus. The result could be helpful to solve the variant problems of information retrieval, information extraction, question answering, and so on.

References

Patrick A. V. Hall and GEOFF R. DOWLING. 1980. Approximate string matching. *Computing Surveys*, 12(4):381–402.

Jun’ichi Kubota, Yukie Shoda, Masahiro Kawai, Hirofumi Tamagawa, and Ryoichi Sugimura. 1993. A method of detecting KATAKANA variants in a document. *IPSJ NL97-16*, pages 111–117.

Sadao Kurohashi and Makoto Nagao. 1999. *Japanese morphological analysis system JUMAN version 3.61.* Department of Informatics, Kyoto University.

Masami Shishibori and Jun’ichi Aoe. 1993. A method for generation and normalization of katakana variant notations. *IPSJ NL94-5*, pages 33–40.

Eiko Yamamoto, Yoshiyuki Takeda, and Kyoji Umemura. 2003. An IR similarity measure which is tolerant for morphological variation. *Natural Language Processing*, 10(1):63–80.