Using Various Features in Machine Learning to Obtain High Levels of Performance for Recognition of Japanese Notational Variants

Masahiro Kojima[a], Masaki Murata[b], Jun'ichi Kazama[c], Kow Kuroda[c], Atsushi Fujita[d,c], Eiji Aramaki[e], Masaaki Tsuchida[c], Yasuhiko Watanabe[a], and Kentaro Torisawa[c]

[a] Dept. of Media Informatics, Ryukoku University, Seta, Otsu-shi, Shiga, 520-2194, Japan
t10m101@mail.ryukoku.ac.jp,watanabe@rins.ryukoku.ac.jp
[b] Dept. of Information and Electronics, Tottori University, Koyama-Minami, Tottori, 680-8550, Japan
murata@ike.tottori-u.ac.jp
[c] National Institute of Information and Communications Technology, Hikaridai, Seika-cho, Kyoto, Japan
{kazama, kuroda, m-tsuchida, torisawa}@nict.go.jp
[d] Faculty of Systems Information Science, Future University Hakodate, Hakodate, 041-8655, Japan
fujita@fun.ac.jp
[e] Center for Knowledge Structuring, University of Tokyo, Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan
eiji.aramaki@gmail.com

Abstract. We proposed a method of using machine learning with various features for the recognition of Japanese notational variants. We increased 0.06 at the F-measure by specific features using existing dictionaries and character pairs useful for recognizing notational variants and obtained 0.91 at the F-measure for the recognition of notational variants. By using the method, we could extract 160 thousand word pairs with a precision rate of 0.9. We also constructed a method using patterns in addition to machine learning and observed that we could extract 4.2 million notational variant pairs with a precision rate of 0.78. We confirmed that our method was much better than an existing method through experiments.

Keywords: machine learning, Japanese notational variant, various features, variant pair, edit distance.

1 Introduction

In Japanese, there exist several different and similar notational variants of a word. For example, "スパゲティ" (*su pa ge te i*) and "スパゲッティ" (*su pa ge tsu te i*) can be used for "spaghetti". A dictionary including such notational variants is extremely useful in many cases such as normalization of word expressions in information retrieval and normalization of word expressions in text mining or information extraction. Therefore, we studied the automatic extraction of the notational variants of a Japanese word.

In terms of related studies, Brill et al. extracted pairs of Japanese Katakana expressions and their corresponding English expressions (Brill *et al.*, 2001), McCallum et al. detected matching or mismatching in restaurant names or paper citation data (McCallum *et al.*, 2005), and Tsuruoka et al., Aramaki et al., and Okazaki et al. extracted synonyms or notational variants in a restricted domain such as biology and medical science (Tsuruoka *et al.*, 2007; Aramaki *et al.*, 2008; Okazaki *et al.*, 2008). In contrast to these studies, we handled the automatic extraction of Japanese notational variants in all the domains.

We used supervised machine learning approaches, in which we can easily use various types of information in the process of extracting Japanese notational variants.

By supervised machine learning approaches (Suzaki *et al.*, 2002; Masuyama and Nakagawa, 2004; Shimada *et al.*, 2005), we used the information that existing dictionaries, including notational variants, provide. We also used the information that similar characters such as "A/a" and "

が/か” (*ga/ka*) are likely to be used for notational variants. By using many other types of useful information, we realized extremely high levels of performance in recognition of Japanese notational variants.

Furthermore, we not only used machine learning but also used pattern based approaches in order to extract a significant number of notational variant pairs (3 million).

In this study, we handled only Japanese notational variants where the edit distance is only one character. This is because many Japanese notational variants satisfy this condition. We think that the extraction of Japanese notational variants where the edit distance is more than one character can be performed by an extended version of the method in the case when the edit distance is one character.

2 Definition of Japanese Notational Variants

Different forms of a word are defined as notational variants. For example, because “スパゲッティ” (*su pa ge tsu te i*) is a different form of “スパゲティ” (*su pa ge te i*), they constitute a pair of notational variants. In the case of “学園闘争” (school conflict) and “学園紛争” (school disturbance), they do not constitute a pair of notational variants because they have the same meaning and synonyms, but “闘争” (conflict) and “紛争” (disturbance) are different words.

3 Difficulty of Extraction of Japanese Notational Variants

Notational variants for a word appear in a similar context. However, we might extract wrong notational variants when only using the condition of a similar context. We thus considered the method of extracting as a pair of notational variants two words that appear in a similar context and where the edit distance is one character. Further, we thought that the method would be able to extract notational variants. We conducted this in the following manner. We used Kazama et al.’s word list, including one million words, each of which has 500 similar words with similarities based on contexts (Kazama and Torisawa, 2008; Kazama *et al.*, 2009). We extracted pairs of similar words where the edit distance is one character from the list and evaluated the results. The F-measure for extracting notational variants was 0.07. We found that using only the condition of a similar context and the condition of the edit distance could not extract notational variants with high performance.

4 Proposed Method

To solve the problems in the previous section and extract notational variants with high performance, we used supervised machine learning methods with various features . We used support vector machines (Cristianini and Shawe-taylor, 2000) as supervised machine learning methods. We used a tool “TinySVM” (Kudoh, 2000) and a linear kernel. We used 1 as the parameter of soft margin. An input of a machine learning method is a pair of words. A machine learning method judges whether or not an input word pair is a pair of notational variants. Most of the features used in our method are described in Table 1.

We also used some other features using heuristic rules similar to F8. F1-F3 are the features related to the characters of different and common parts that were often used in existing studies. The other features are the newly made ones to obtain high levels of performance. F4 was given by Kazama et al.’s word list. F5-F7 used the characteristic that a word pair whose different parts are similar characters (e.g., “３/三” (three), “A/a,” and “あ/ぁ”) is likely to be a notational variant pair. F8 was a heuristic rule. F9-F12 were based on dictionaries. F12 used a stacking algorithm (van Halteren *et al.*, 2001). In F12, we first made a training data set comprising 25,934 data items (positive examples), which were notational variant pairs, and 904,612 data items (negative examples), which were not notational variant pairs, by using the JUMAN dictionary (Kurohashi and Kawahara, 2009), which has information on notational variants. An input data item was judged as a positive or negative example using this training data set and the judged result was given as F12.

Table 1: Features

ID	Definition
F1	A character of the first or second word in a different part
F2	1 to 3 grams characters before or after a different part
F3	Types of characters (Hiragana, Katakana, Chinese Kanaji characters, and symbols), parts of speeches (POS) of a morpheme including the character,and the location in the morpheme for F1 and F2
F4	The similarity between the two input words
F5	Whether or not the characters of the first and second words in a different part indicate the same number when they are numbers. (e.g., “一週間/1週間”(a week))
F6	Whether or not the characters of the first and second words in a different part are the same characters, one of which is in the upper case and the other is in the lower case (e.g., “おかぁちゃん”(*o ka a' cha n*) and “おかあちゃん”(*o ka a cha n*)), or they are similar characters (e.g., “が/か” (*ga/ka*)) when they are Hiragana/Katakana characters
F7	Whether or not the characters of the first and second words in a different part are the same characters, one of which is in the upper case and the other is in the lower case (e.g., “300kbps” and “300Kbps”) when they are alphabets
F8	Whether or not one of the characters of the first and second words in a different part is the word indicating a digit or a figure (e.g., “億” (million), “千” (thousand)) and the other is missing
F9	Whether or not parts, including different parts, are recognized as a notational variant pair by the JUMAN dictionary
F10	Whether or not parts, including different parts, have the same reading or sound recognized using the JUMAN dictionary
F11	Whether or not the characters of the first and second words in a different part are Chinese characters and heterothallic (*Itaiji*) Chinese characters (e.g., “沢” (*sa wa*) and “澤” (*sa wa*), which are originally the same characters.)
F12	Whether or not a stacking algorithm judges a word pair as a notational variant pair based on the JUMAN dictionary

5 Experiments

5.1 Data Sets Used in Experiments

We constructed data sets used in experiments by using Kazama et al.'s word list, including one million words, each of which has 500 similar words. We randomly extracted 14,185 pairs of similar words where the edit distance is one character from the list. Three annotators manually tagged the tags that indicate whether data items are notational variants or not to them, and we used the results by their voting for experiments. We obtained a high kappa value of 0.84 in tagging agreement (Landis and Koch, 1977). We made from them word pairs where the first word and the second word in a pair were exchanged. Thus, we made a total of 28,370 word pairs. We evenly divided 28,370 word pairs into two parts: Data set A and Data set B. Data set A included 745 pairs of notational variants. Data set B included 725 pairs of notational variants.

We used Data set A for considering new types of features. We used Data set B for confirming that the new types of features are useful. We considered new types of features in experimental results of the cross validation using Data set A. We call this experiment as Experiment A. After we determined the features used in machine learning, we conducted experiments where Data set A was used as a training data set and Data set B was used as a test data set. We call this experiment as Experiment B.

5.2 Experimental Results

We conducted Experiment B. The results are shown in Table 2.

The “proposed method” was our method using all the features in Table 1 described in Section 4. In the “comparison method A,” all the input data items were judged as pairs of notational variants. This is the same as the results described in Section 3. In “comparison method B,” only the general

Table 2: Experimental results

Method	Recall	Precision	F-measure
Proposed method	0.90	0.93	0.91
Comparison method A	1.00	0.04	0.07
Comparison method B	0.81	0.89	0.85
Comparison method C	0.58	0.79	0.67
Comparison method D	0.24	0.02	0.04
Comparison methods E/F	0.30	0.97	0.45

Table 3: The similar features as those in Aramaki *et al.*'s studies

ID	Definition
F'1	A character of the first or second word in a different part
F'2	Previous character of F'1
F'3	Subsequent character of F'1
F'4	Types of characters (Hiragana, Katakana, and Chinese Kanaji characters, and symbols) for F'1
F'5	Types of characters (Hiragana, Katakana, and Chinese Kanaji characters, and symbols) for F'2
F'6	Types of characters (Hiragana, Katakana, and Chinese Kanaji characters, and symbols) for F'3
F'7	This feature (edit distance-based similarity:SIM_{ed}) between word pair(w_1, w_2) is defined as follows: $SIM_{ed}(w_1, w_2) = 1 - \frac{EditDistance(w_1, w_2) \times 2}{len(w_1) + len(w_2)}$, where $len(w_1)$ is the number of characters of w_1, $len(w_2)$ is the number of characters of w_2, $Editdistance(w_1, w_2)$ is the minimum number of point mutations required to change w_1 into w_2. For details, see(Levenshtein, 1965)

features were used (F1-F3). "Comparison methods C and D" were similar to those used in Aramaki et al. (Aramaki *et al.*, 2008). In "comparison method C," SVM with the similar features as those in Aramaki et al.'s studies was used by using Data set A, which was used as a training data set, and Data set B, which was used as a test data set. The features used in "comparison method C" are shown in Table 3. "Comparison method D" indicated the results when using Aramaki et al.'s tool.[1] "Comparison method E" was the method where only data items that had F5, F7, or F10 were judged to be pairs of notational variants. "Comparison method F" was a method based on SVM using only F5, F7, and F10 as features. F5, F7, and F10 were features that were found to be especially useful in the experiments described in a later section.

We confirmed that the "proposed method" obtained a significantly higher F-measure than all the other methods in Table 2 at the significance level of 0.05. In this paper, we used a bootstrapping method (Efron, 2001) for statistical significant differences. In the bootstrapping method, we assume that we want to compare Methods 1 and 2. We randomly and redundantly extract N data items from an evaluated data set. N is the number of data items in an evaluated data set. We repeat this 10,000 times and obtain 10,000 data sets. We obtain the F-measures of Methods 1 and 2 for 10,000 experiments using 10,000 data sets. In 10,000 experiments, we calculate the ratio in which Method 1 obtains higher F-measures than Method 2. When the ratio is larger than 0.95, we can roughly estimate that Method 1 is better than Method 2 at the significance level of 0.05.

The results indicated the following: i) Our proposed method obtained a highly accurate F-measure (0.91). ii) The F-measure obtained by our proposed method was 0.06 higher than "comparison method B." This indicates that using F4-12 features in addition is extremely important. iii) Our proposed method obtained a higher F-measure than "Comparison methods C/D" based on Aramaki et al. studies. "Comparison method D" often made errors when the difference or surrounding parts included numbers. We think that Aramaki et al. would use a training data set in a restricted domain and not use a training data set that includes numbers.

[1] http://202.218.239.69/~aramaki/TRANS/

Table 4: Changes in F-measure when eliminating only one feature

Eliminated feature	Experiment A	Experiment B	Eliminated feature	Experiment A	Experiment B
F1	-0.002	-0.002	F7	-0.015	-0.014
F2	0.000	-0.002	F8	-0.001	0.001
F3	-0.007	0.001	F9	0.001	0.000
F4	0.000	-0.001	F10	-0.020	-0.052
F5	-0.027	-0.012	F11	0.000	0.000
F6	0.001	0.001	F12	-0.003	-0.004

Table 5: Feature analysis using a bootstraping method

Eliminated feature	Eliminating only one feature and using all the remaining features		Using all the features	
	Experiment A	Experiment B	Experiment A	Experiment B
F1	0.1584	0.0000	0.7994	0.9836
F2	0.8388	0.1258	0.1573	0.8637
F3	0.0150	0.5524	0.9850	0.4473
F4	0.4848	0.2105	0.4967	0.7460
F5	0.0000	0.0044	1.0000	0.9956
F6	0.6357	0.5720	0.0000	0.4212
F7	0.0000	0.0000	1.0000	1.0000
F8	0.2441	0.6315	0.7084	0.0000
F9	0.5776	0.0000	0.3629	0.0000
F10	0.0000	0.0000	1.0000	1.0000
F11	0.3416	0.0000	0.5215	0.6413
F12	0.1973	0.0385	0.7951	0.9556

5.3 Examining Features

In order to examine which features are important, we compared the results using all the features and the results eliminating only one feature and using all the remaining features. We conducted these experiments in Experiments A and B. We used a bootstrapping method to compare the two methods. The results are shown in Table 4. Table 4 indicates the ratios when the method eliminating only one feature and using all the remaining features obtained a higher F-measure than the method using all the features and the ratios when the method using all the features obtained a higher F-measure than the method eliminating only one feature and using all the remaining features. Table 5 indicates the changes of F-measures when eliminating only one feature and using all the remaining features against the case when using all the features. Table 5 shows that F5, F7, and F10 were the features where the elimination of each feature was significantly worse than the use of all the features in both Experiments A and B at the significance level of 0.05 by a bootstrapping method. The values of significance level of 0.05 by a bootstrapping method are underlined in Table 5. Table 4 shows that no use of F5, F7, or F10 decreased (0.027/0.012), (0.015/0.014), or (0.020/0.052) at the F-measure in both Experiments A and B. We found that these features were highly important.

We thus conducted experiments using comparison methods E/F in Section 5.2. From Table 2, we found that although F5, F7, and F10 were important, the use of F5, F7, and F10 only was not good.

We used a stacking algorithm in this study (F12). In our experiments, we found that the use of F12 improved (0.003/0.004) at the F-measure and the elimination of the feature was significantly worse than the use of all the features in both Experiments A and B at the significance level of 0.1 by a bootstrapping method. We found that F12 was also useful.

In our experiments, we found that there were no features where the elimination of each of the features was significantly better than the use of all the features in both Experiments A and B at the

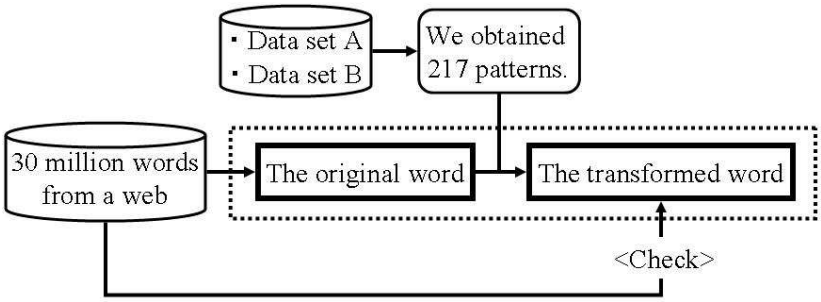

Figure 1: The outline of our pattern-based method extraction notational variants

Table 6: Acquired patterns

Acquired pattern	Notational variant pairs
"・/delete"	"塩・こしょう/塩こしょう" (*sio ・ko syo u/sio ko syo u*) (salt and pepper)
"ル (*ru*) /ー"	"ハルマゲドン/ハーマゲドン" (*ha ru ma ge do n/ha a ma ge do n*) (Armageddon)
"篭/ご"(*ko*)	"引き篭もり/引ぎこもり (*hi ki ko mo ri/hi ki ko mo ri*) (withdrawal)

significance level of 0.05 by a bootstrapping method. Therefore, we found that all the features we used in this study were useful and we had better use all the features.

5.4 Extracting Notational Variants

By using all the word pairs in Kazama et al.'s word list as input data items for our method, we obtained 160 thousand word pairs as notational variants. Because the precision rate of our method is about 0.9 (Table 2), the extracted pairs would include 145 ($= 160 \times 0.9$) thousand correct notational variant pairs. Because the Juman dictionary includes only 20 thousand notational variant pairs, our results have a significant impact.

6 Pattern-based Extraction of a Very Large Number of Notational Variants

We conducted experiments extracting a large number of notational variants by using patterns in addition to machine learning. Figure 1 shows the outline of this method.

In the experiments, we first constructed patterns that are useful for making notational variants. We extracted different parts (e.g., "６/六" (six)) of notational variant pairs (e.g., "６メートル/６メートル" (six meters)) as such patterns from Data sets A and B. We obtained 217 patterns. Some of the acquired patterns are described in Table 6.

Next, we extracted 30 million words from a web (Shinzato *et al.*, 2008). We transformed each of the 30 million words (the original word; e.g., "６キログラム" (six kilograms)) to the other word (the transformed word; e.g., "六キログラム") by using patterns. When the transformed word was included in the 30 million words, we made a pair of the original and transformed words as a candidate of a notational variant pair. We performed the experiments and obtained 8.3 million candidates.

Finally, we applied our method extracting a notational variant pair described in Section 4. As a result, we obtained 4.2 million word pairs as notational variant pairs. We randomly extracted 300 pairs and evaluated them. We found that 233 pairs among the 300 pairs (78%) were correct.
Therefore, the extracted pairs would include 3.3 ($= 4.2 \times 0.78$) million correct notational variant pairs.

Some of the extracted notational variant pairs are described in Table 7. Different parts are underlined.

Table 7: Extracted notational variant pairs

ホーエンツォレルン家 (*ho - e n tsu o' re ru n ke*) (House of Hohenzollern)	ホーエンツオレルン家 (*ho - e n tsu o re ru n ke*)
青山1丁目駅 (*ao yama 1 cho me eki*) (the station of the 1st street in Aoyama a station name)	青山一丁目駅 (*ao yama ichi cho me eki*)
ラムズフェルド国防長官 (*ra mu zu fu e ru do koku bou cho kan*) (Secretary of Defense Rumsfeld)	ラムスフェルド国防長官 (*ra mu su fu e ru do koku bou cho kan*)
カリフォルニア州法 (*ka ri fo ru ni a shu hou*) (the law of the state of California)	カルフォルニア州法 (*ka ru fo ru ni a shu hou*)
敷金・礼金無し (*shiki kin ・ rei kin na shi*) (no security deposit and no key money)	敷金・礼金なし
PukiWIkiMod	PukiWikiMod

7 Conclusion

We proposed our method of using machine learning with various features for the recognition of Japanese notational variants. We increased 0.06 at the F-measure by specific features using existing dictionaries and character pairs useful for recognizing notational variants and obtained 0.91 at the F-measure for the recognition of notational variants. We confirmed that our method was much better than an existing method through experiments. We obtained the results where we could extract 160 thousand word pairs with a precision rate of 0.9.

We found that the features (F5, F7, and F10) that used numbers (Arabic and Chinese numerals), alphabets (low and upper cases), and word reading were especially useful in our experiments. The feature (F12) using a stacking algorithm was also useful.

We also conducted experiments where we constructed a large number of notational variant pairs by using patterns in addition to machine learning. From the experiments,we would be able to extract 4.2 million notational variant pairs with a precision rate of 0.78.

We have a plan to release a database on the extracted notational variants that are checked and corrected by hands in this year.

References

Aramaki Eiji, Takeshi Imai, Kengo Miyo, and Kazuhiko Ohe. 2008. Orthographic disambiguation incorporating transliterated probability. *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp.48-55.

Brill Eric, Gary Kacmarcik, and Chris Brockett. 2001. Automatically harvesting katakana-english term pairs from search engine query logs. *Proceedings of Sixth Natural Language Processing Pacific Rim Symposium*, pp.393-399.

Cristianini Nello and John Shawe-Taylor. 2000. An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press.

Dagan Ido, Fernando Pereira, and Lillian Lee. 1994. Similarity-based estimation of word cooccurrence probabilities. *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pp.272-278.

Efron Bradley. 1979. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1),1-26.

Kazama Jun'ichi and Kentaro Torisawa. 2008. Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. *Proceedings of the 46th annual meeting on Association for Computational Linguistics*, pp.407-415.

Kazama Jun'ichi, Stijn De Saeger, Kentaro Torisawa,and Masaki Murata. 2009. Generating a large-scale analogy list using a probabilistic clustering based on noun-verb dependency profiles. *15th Annual Meeting of The Association for Natural Language Processing*, pp.84-87. (in Japanese).

Knight Kevin and Jonathan Graehl 1997. Machine transliteration. *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pp.128-135.

Kudoh Taku. 2000. TinySVM: Support vector machines. http://cl.aist-nara.ac.jp/taku-ku//software/TinySVM/index.html.

Kurohashi Sadao and Daisuke Kawahara. 2009. Japanese Morphological Analysis System JUMAN version 6.0. Department of Informatics, Kyoto University.

Landis J. Richard and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, pp.159-174.

Levenshtein Vladimir Iosifovich. 1965. Binary codes capable of correcting deletions, insertions and reversals. *Doklady Akademii Nauk SSSR*, 163(4),845-848.

Masuyama Takeshi and Hiroshi Nakagawa. 2004. Two step POS selection for SVM based text categorization. *IEICE Transactions on Information and Systems*, 87-D(2),373-379.

McCallum Andrew, Kedar Bellare, and Fernando C. N. Pereira. 2005. A conditional random field for discriminatively-trained finitestate string edit distance. *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, pp.388-395.

Okazaki Naoaki, Yoshimasa Tsuruoka, Sophia Ananiadou, and Jun ' ichi Tsujii. 2008. A discriminative candidate generator for string transformations. *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp.447-456.

Shimada Kazutaka, Koji Hayashi, and Tsutomu Endo. 2005. Product specification extraction using SVM and transductive SVM. *Journal of Natural Language Processing*, 12(3),43-66.

Shinzato Keiji, Tomohide Shibata, Daisuke Kawahara, Chikara Hashimoto, and Sadao Kurohashi. 2008. Tsubaki: An open-search engine infrastructure for developing new information access. *The Third International Joint Conference on Natural Language Processing*, pp.189-196.

Suzuki Jun, Yutaka Sasaki and, Eisaku Maeda. 2002. SVM Answer Selection for Open-Domain Question Answering. *Proceedings of the 19th International Conference on Computational Linguistics*, pp.974-980.

Tsuruoka Yoshimasa, John McNaught, Jun'ichi Tsujii, and Sophia Ananiadou. 2007. Learning string similarity measures for gene/protein name dictionary look-up using logistic regression. *Bioinformatics*, 23(20),2768-2774.

van Halteren Hans, Jakub Zavrel, and Walter Daelemans. 2001. Improving accuracy in word class tagging through the combination of machine learning systems. *Computational Linguistics*, 27(2),199-229.